Positionnement visuel pour la réalité augmentée en environnement plan

Gilles Simon

UNIVERSITÉ DE LORRAINE

École doctorale IAEM Lorraine

Positionnement Visuel pour la Réalité Augmentée en Environnement Plan

Mémoire

présenté et soutenu publiquement le 9 décembre 2019

pour l'obtention d'une

Habilitation à Diriger des Recherches

(mention informatique)

par

Gilles Simon

Composition du jury

Rapporteurs :	Mme Agnès Desolneux, Directrice de Recherche, CNRS et Professeur, ENS M. Eric Marchand, Professeur, Univ. Rennes 1, IRISA M. Peter Sturm, Directeur de Recherche, Inria Grenoble Rhône-Alpes
Examinateurs :	Mme Marie-Odile Berger, Directrice de Recherche, Inria Nancy - Grand Est M. Cédric Demonceaux, Professeur, Univ. Bourgogne, Le2i M. Antoine Tabbone, Professeur, Univ. Lorraine, Loria

Laboratoire Lorrain de Recherche en Informatique et ses Applications — UMR 7503

Loria
Laboratoire lorrain de recherche en informatique et ses applications

À Maria del Carmen

Avant-propos

Ce mémoire synthétise mes travaux de recherche sur le positionnement visuel en environnement bâti, réalisés au sein de l'équipe Magrit [1] durant la période qui sépare le début de la thèse de Javier Flavio Vigueras Gomez, en 2003, de la fin de la thèse d'Antoine Fond, en 2018.

J'ai fait le choix de présenter ces travaux dans un ordre non chronologique et de manière plus ou moins détaillée selon leur ancienneté. Ainsi, si les chapitres 2, 3 et 4 décrivent assez précisément des travaux datant de 2018, 2017 et 2018, respectivement, le chapitre 5 présente de manière plus condensée des contributions allant de 2003 à 2013. J'ai volontairement écarté de ce mémoire certains travaux, plus en marge du positionnement visuel (reconnaissance de lieux [21], influence d'erreurs de calibration de la caméra sur le calcul de pose [26, 25] etc.) ou relatifs au positionnement visuel appliqué à d'autres domaines (didactique des sciences [54, 18, 16] et environnements industriels [40, 13, 50]).

Concernant les citations, nous faisons référence à nos propres travaux par de simples nombres entre crochets (exemple : [17]), renvoyant à la rubrique "Publications de l'auteur", page 145, et aux travaux des autres auteurs par leurs initiales suivies de l'année de publication et parfois d'une lettre permettant de distinguer des abréviations identiques (exemple : [RD06b]), renvoyant à la rubrique "Bibliographie", page 135. Je fais aussi parfois référence à des vidéos, que l'on pourra trouver sur mon site internet à l'adresse `https://members.loria.fr/GSimon/habilitation-a-diriger-des-recherches/`.

Il appartient enfin à cet avant-propos de préciser que les poèmes et schémas (que j'ai essayé de reproduire au mieux) placés en exergue de chaque chapitre sont d'Eugène Guillevic (1907-1997), poète et écrivain français ayant habité à Ferrette, charmant village du Sundgau alsacien où vécurent aussi mes grand-parents maternels. Ils sont extraits du recueil intitulé *Euclidiennes*, publié en 1967 et toujours disponible dans la collection NRF *Poésie*/Gallimard. Ces poèmes ont pour titre *point* (page 27), *rectangle* (page 51), *figures* (page 71), *plan* (page 89) et *pyramide* (page 109 – je n'ai repris que les quatre derniers vers de ce poème).

Liverdun, le 30 mai 2019.

1. Équipe mixte Inria - Université de Lorraine de l'UMR Loria (Laboratoire lorrain de recherche en informatique et ses applications), dirigée par Marie-Odile Berger, DR Inria.

Table des matières

Chapitre 1

Introduction

1.1 Sujet du mémoire

Commençons par préciser les termes et le périmètre du problème dont traite ce mémoire : "Positionnement visuel pour la réalité augmentée en environnement plan".

1.1.1 Positionnement

Le positionnement est défini comme l'action de déterminer la position d'un objet. Par exemple, le GPS (*Global Positioning System*) est un système de positionnement par satellites. Nous utilisons toutefois une définition étendue de ce terme, ajoutant la détermination de l'orientation de l'objet à celle de sa position. Nous emploierons fréquemment l'anglicisme *pose* pour désigner de façon concise la position et l'orientation d'un objet. La pose est évidemment toujours exprimée dans un référentiel prédéfini. Par exemple, le GPS permet de localiser un récepteur dans un repère terrestre (latitude, longitude, altitude), tout comme un magnétomètre permet de mesurer une orientation par rapport à un point cardinal.

1.1.2 Visuel

Le positionnement en milieu urbain possède des applications importantes dans les domaines de la géomatique, de la navigation assistée ou autonome, de la robotique et de la réalité augmentée. Le GPS est le capteur les plus utilisé dans ces applications. Malheureusement, dans les milieux urbains, la présence d'immeubles de part et d'autre de la route peut obstruer le signal satellite, ce qui dégrade la précision de ce système (effet de vallée). Même non dégradée, avec des erreurs qui varient de 5 à 10 mètres, la mesure GPS est trop imprécise pour des applications de réalité augmentée ou de robotique mobile performantes. De nombreuses recherche ont été menées pour tenter d'améliorer la précision du positionnement en milieu urbain à l'aide de la vision par ordinateur. Le positionnement dit *visuel* ne requiert d'autre capteur qu'une caméra monoculaire. Il permet de mesurer la pose de la caméra elle-même ou d'un objet (par exemple, des lunettes de réalité augmentée) solidaire de la caméra. Il repose traditionnellement sur la connaissance d'un modèle 3D de la scène pouvant prendre différentes formes (nuage de points, modèle polyédrique etc.) associé à des balises visuelles (qui elles aussi peuvent être de natures très différentes, voir ci-dessous). La pose est alors obtenue dans le repère du modèle 3D comme solution au problème d'alignement entre les balises projetées dans le plan image (selon la perspective centrale) et les balises détectées.

1.1.3 Pour la réalité augmentée

La réalité augmentée (RA) est un des thèmes centraux de l'équipe Magrit. Nous avons très tôt tenté de l'appliquer aux environnements architecturaux, en raison notamment de nos relations privilégiées avec le CRAI, le Centre de Recherche en Architecture et Ingénierie de Nancy. Nous avons ainsi, dès 1996, travaillé en collaboration avec le CRAI et EDF, sur l'éclairage par

synthèse du pont Neuf, en lien avec son environnement. À cette époque le temps réel était difficilement atteignable, mais nous étions parvenus, en utilisant une méthode de recalage 3D-2D robuste [37, 57] (voir la vidéo 1), à incruster un projet d'illumination du pont Neuf dans une vidéo, en tenant compte des reflets de l'éclairage (virtuel) sur l'eau (réelle) et de ses occultations par les objets (réels) de la scène [39] (voir la vidéo 2). L'étude d'impact de projets architecturaux ou d'aménagement urbain est un des champs d'application les plus importants de la RA, mais d'autres applications ont vu le jour dans les environnements urbains. Le secteur du tourisme peut par exemple bénéficier de l'affichage d'anecdotes historiques sur les façades des bâtiments à visiter [LKT17]. Des informations en surimpression de l'image peuvent également profiter à la publicité dans les quartiers marchands. Si ce type d'information ne nécessite qu'un positionnement relatif une fois le bâtiment identifié, d'autres applications ont besoin d'un positionnement global géoréférencé. Ainsi, on peut également penser à des annotations visuelles (flèches ou chemin tracé au sol) pour aider à trouver sa route vers une destination spécifique dans une ville inconnue ou mieux comprendre un réseau de transport public [KG11]. Le domaine de la maintenance des infrastructures souterraines peut également profiter d'applications en RA pour les services de voirie en incorporant par exemple à l'image les plans des réseaux souterrains (électricité, gaz) [KAAA13]. Le secteur du divertissement a également déjà montré son intérêt pour la RA urbaine avec des succès vidéo-ludiques tels que *Ingress* ou *Pokémon Go*. À noter que les méthodes proposées dans ce mémoire s'appliquent aussi bien à des environnements urbains réels qu'à des reproductions miniatures. Cela permet d'envisager des tâches de conception ou de planification réalisées par des architectes ou des urbanistes interagissant autour d'une maquette à l'aide de la RA [AZG+19].

Le positionnement pour la RA possède ses propres spécificités. D'une part, les objets virtuels à intégrer en temps réel au flux d'images vidéo doivent être préalablement positionnés dans le même repère 3D que celui dans lequel la pose de la caméra est calculée. Dans le cas d'un positionnement par GPS et magnétomètre[1], les objets à ajouter doivent être géoréférencés. Dans le cas d'un positionnement visuel, ils doivent être positionnés par rapport au modèle 3D dans lequel les balises visuelles ont été définies. Cela implique par exemple que les méthodes de SLAM visuel (*Simultaneous Localization and Mapping*) [DM02], qui permettent de calculer à la volée une carte de l'environnement (le plus souvent, un nuage de points), en même temps que la pose de la caméra par rapport à cette carte, sont rarement directement utiles à la RA (les objets virtuels ne peuvant être positionnés dans le modèle 3D, non connu *a priori*), alors qu'elles le sont à la robotique, en permettant par exemple à un robot de se déplacer de manière autonome dans son environnement, tout en évitant les obstacles.

D'autre part, la précision attendue d'un système de positionnement n'est pas identique selon que l'on se place dans un contexte de RA ou dans un autre contexte. Le GPS est par exemple suffisamment précis pour guider sans ambiguïté un conducteur automobile vers sa destination. En revanche, il est trop imprécis pour avoir l'illusion d'ancrage de la scène virtuelle à la scène réelle dans une application de RA (voir par exemple la vidéo 3, dans laquelle les incrustations reposent uniquement sur des données capteurs – GPS + magnétomètre : un effet de tremblement de la scène virtuelle est observé). En RA, la précision obtenue sur la position est peu informative, ce qui importe surtout c'est de se rendre compte visuellement de l'impact de cette précision sur l'alignement entre les éléments réels et virtuels des images augmentées. En ce sens, le positionnement visuel permet, dans certaines conditions, d'obtenir un excellent ancrage du point de vue perceptif. En revanche, le positionnement visuel est réputé pour être plus instable que le positionnement basé capteurs. Dans ses déclinaisons traditionnelles, la précision de la pose obtenue dépend en effet essentiellement du nombre de balises correctement identifiées dans les images (et aussi de leur répartition spatiale). Or cette identification peut être perturbée par

1. Un magnétomètre ne permet pas seul de retrouver les trois angles de rotation (dits angles d'Euler) d'une caméra. Mais certaines centrales inertielles (IMU – *Inertial Measurement Unit*) intègrent un magnétomètre, ce qui permet d'obtenir trois angles (dans un repère de directions Nord, Est et le vecteur gravité) à l'aide d'un filtre de Kalman. J'utilise le terme "magnétomètre" au lieu de "IMU équipée d'un magnétomètre" pour éviter des phrases trop lourdes ainsi que toute confusion avec l'utilisation des termes "IMU" et "centrale inertielle", qui permettent d'obtenir des mesures d'odométrie aussi bien sur la position que l'orientation.

différents facteurs contextuels tels que les conditions météorologiques, l'alternance jour/nuit, la présence de piétons ou de véhicules devant les balises et le caractère plus ou moins distinctif de ces balises. Dans certains cas, le nombre de balises identifié est trop faible pour obtenir une pose suffisamment précise du point de vue ancrage, voire pour être en mesure de calculer la pose. Cela peut avoir pour conséquences des interruptions de l'affichage de la scène virtuelle ou un effet de tremblement plus ou moins prononcé au niveau de cet affichage. Nos travaux ont pour visée d'augmenter la stabilité du positionnement visuel en environnement urbain, tout en conservant une précision suffisante pour la RA.

1.1.4 En environnement plan

Nous utilisons le terme "environnement plan" plutôt que, par exemple, "environnement urbain" pour deux raisons. D'une part, pour insister sur le fait que nous nous appuyons uniquement sur les objets pérennes des environnements urbains, constitués essentiellement du bâti. Dans les environnements urbains, des véhicules apparaissent aussi vite qu'ils disparaissent, des arbres poussent et perdent leurs feuille en automne, des auvents, parasols, tables et chaises sont sortis lorsqu'il fait beau et rentrés lorsqu'il pleut ou qu'il fait nuit, etc. Une grande partie du bâti en revanche, peut être considérée comme pérenne et nous utilisons en particulier les façades de bâtiments comme balises visuelles. D'autre part, parce ce que ce terme englobe les scènes d'intérieur aussi bien que d'extérieur. Nous nous sommes en effet intéressés à l'aide à l'ameublement *via* la RA dans le cadre d'un projet européen dont les partenaires industriels étaient Intracom et Ikea Grèce [44, 5]. Le point commun entre les environnements intérieurs et extérieurs est qu'ils contiennent des murs, c'est-à-dire des surfaces planes qui présentent des propriétés intéressantes que nous exploitons dans l'ensemble des travaux présentés ici. Aussi les travaux décrits aux chapitres 2 et 5 sont-ils valables aussi bien pour des scènes d'intérieur que d'extérieur. En revanche, les travaux décrits aux chapitres 3 et 4 exploitent les arrangements spatiaux entre des éléments architecturaux d'une façade (fenêtres, balcons etc.), non visibles dans les scènes d'intérieur.

1.2 État de l'art général

Le positionnement en environnement plan a fait l'objet de nombreux travaux en vision par ordinateur, dont j'ai choisi de retenir les plus emblématiques, représentatifs de catégories de méthodes auxquelles d'autres travaux, plus anciens ou moins performants, peuvent être rattachés sans toutefois être mentionnés dans cet état de l'art. Comme catégories principales je distinguerai les méthodes purement visuelles des méthodes nécessitant de connaître une estimée initiale plus ou moins précise de la pose, à l'aide de capteurs externes.

1.2.1 Méthodes purement visuelles

Les méthodes de la première catégorie peuvent être regroupées en trois paquets : celles qui considèrent des descripteurs locaux de points détectés dans l'image à traiter (appelée *image requête* par la suite), celles qui considèrent un descripteur global de cette image et enfin, celles qui infèrent directement la pose à partir de l'image en utilisant un réseau de neurones convolutifs (CNN ou ConvNet pour *Convolutional Neural Networks*).

1.2.1.1 Utilisation de descripteurs locaux

Une technique couramment employée consiste à reconstruire, préalablement à l'utilisation du système, un nuage de points 3D de l'environnement à l'aide d'une technique de SFM (*Structure from Motion* [FZ98]) ou de SLAM (*Simultaneous Localization and Mapping*), et à calculer les poses à partir d'appariements entre des descripteurs de type SIFT [Low99] associés d'une part aux points du nuage 3D et d'autre part à des points détectés dans les images du flux vidéo. L'algorithme RANSAC [FB81] couplé à la méthode P3P [XXJH03] sont généralement utilisés pour un calcul robuste de pose à partir des correspondances 3D-2D. Cette technique est sans

doute la plus ancienne, mais des travaux sont toujours en cours notamment pour permettre un appariement efficace entre les descripteurs lorsque le modèle contient un très grand nombre de points 3D, comme cela est le cas à l'échelle d'une ville (plusieurs millions de descripteurs sont généralement à considérer).

Une des méthodes les plus efficientes actuellement est nommée *Active Search* [SLK17]. Cette méthode accélère l'appariement des descripteurs à l'aide d'une recherche priorisée. Elle utilise un vocabulaire visuel (ou sac de mots visuels – BoW pour *Bag of Words*) pour quantifier l'espace des descripteurs au moyen d'un partitionnement de type *k-means* [PCI+07]. Dans une étape préliminaire, chaque descripteur de point 3D est affecté au mot visuel (à la partition) le plus proche. Lors d'un calcul de pose, Active Search compte, pour chaque descripteur de l'image requête, le nombre de descripteurs de points 3D affectés au même mot visuel que ce descripteur. Cela détermine le nombre de comparaisons à effectuer pour apparier ce descripteur. Les descripteurs de l'image requête sont alors appariés (en utilisant le critère de Lowe [Low99]) par ordre croissant du nombre de comparaisons à effectuer. Si un appariement 2D-3D est trouvé, la méthode tente de trouver des correspondances 3D-2D supplémentaires avec les points 3D entourant le point apparié. La recherche de correspondances s'arrête lorsque 100 correspondances ont été obtenues. Cette méthode parvient à réaliser l'étape d'appariement en un temps moyen allant de 0,10 s pour un modèle comprenant 1,65 millions de points à 0,97 s pour un modèle en comprenant 36,15 millions [2] [SMT+18].

La répétabilité des descripteurs locaux tels que SIFT est cependant peu robustes aux changements d'illumination et aux forts changements de point de vue. Une version "apprise" de SIFT, appelée LIFT (*Learned Invariant Feature Transform*) a été proposée en 2016 [YTFLF16]. Le détecteur de points, l'orientation des patchs [3] ainsi que les descripteurs sont obtenus à l'aide de trois CNN entraînés à partir de solutions de SFM obtenues dans divers environnements. Les points et descripteurs obtenus par les algorithmes de SFM utilisés sont des primitives SIFT. On peut donc dire que LIFT "apprend SIFT". Toutefois, divers benchmarks [SHSP17] font apparaître de légères différences entre les deux méthodes : si LIFT obtient globalement de meilleurs taux de rappel (nombre d'*inliers* [4] du RANSAC / nombre de correspondances correctes), SIFT a de meilleurs taux de précision (nombre d'*inliers* / nombre de correspondances avant RANSAC). LIFT fait par ailleurs preuve d'une plus grande robustesse au flou, à la compression JPEG et aux changements d'exposition, mais SIFT s'avère plus robuste (bien que faiblement robuste [SMT+18]) à l'alternance jour/nuit, ainsi qu'aux rotations de l'image. Par ailleurs, LIFT est moins performant, sur la quasi-totalité des benchmarks, que SIFT-PCA [BTJ15] et DSP-SIFT [DS15], deux variantes de SIFT [5].

En plus des problèmes de robustesse à divers facteurs contextuels, les descripteurs locaux souffrent d'un manque de discrimination comme le révèlent les taux de précision obtenus par SIFT et LIFT dans les benchmarks présentés dans [SHSP17] (de l'ordre de 40% entre deux images issues d'une même vidéo, ou provenant d'images Internet de sources différentes). Les patchs sur lesquels repose leur calcul ont en effet une taille limitée (dépendant de l'échelle du point), finalement assez pauvre en information. Il est ainsi fréquent de détecter des patchs similaires (à une orientation et échelle près, ce qui donne lieu à des descripteurs identiques) autour de points physiquement différents (par exemples, deux coins d'une même fenêtre) ou même sémantiquement différents (par exemple, un coin de fenêtre et un coin de porte).

Enfin, si des méthodes telles que Active Search permettent d'apparier les descripteurs relativement rapidement malgré le nombre important de descripteurs à considérer, il n'en reste pas moins que les modèles pris en compte sont très lourds en mémoire (plusieurs gigaoctets pour

2. Sur un PC équipé d'un processeur Intel Core i7-4770 CPU avec 3.4GHz, 32Go de RAM, et une carte NVidia GeForce GTX 780 GPU

3. Nous conservons ce terme anglais pour désigner les "imagettes" ou sous-fenêtres de l'image de tailles variables centrées autour des points détectés.

4. Points appartenant à l'ensemble de consensus détecté par l'algorithme RANSAC, en opposition aux points *outliers*, rejetés par l'algorithme.

5. SIFT-PCA [BTJ15] utilise l'Analyse en Composantes Principales – ACP, pour projeter les descripteurs SIFT dans un espace de plus petite dimension ; DSP-SIFT [DS15] calcule les gradients à de multiples échelles, au lieu de les calculer uniquement à l'échelle à laquelle la primitive SIFT a été détectée.

Dubrovnik [LSH10]), ce qui peut être un frein à leur exploitation sur des dispositifs légers tels que les smartphones.

1.2.1.2 Utilisation d'un descripteur global

Une autre approche consiste à utiliser un BoW calculé sur des descripteurs (toujours de type SIFT) obtenus dans une base d'images représentant diverses vues de l'environnement et associées à des poses connues. Un tel modèle peut bien sûr être obtenu à l'aide d'un algorithme de SFM ou de SLAM. La différence avec les méthodes précédentes est que chaque image de la base, tout comme l'image requête, sont décrites par un seul vecteur correspondant à l'histogramme des mots visuels (partitions les plus proches des descripteurs) détectés dans ces images. La pose correspondant à l'image requête est approximée par la pose correspondant à l'image la plus proche, au sens de ce descripteur, dans la base de données. La méthode présentée dans [TAS+15] implémente ce schéma, à l'aide d'une extension des BoW appelée VLAD (*Vector of Locally Aggregated Descriptors*) [JDSP10]. Le descripteur global VLAD utilise également un BoW, mais incorpore les résidus de chaque descripteur local par rapport au mot visuel auquel il est assigné, ce qui le rend généralement plus discriminant qu'un descripteur global classique.

La recherche de la vue la plus proche est très rapide, puisqu'elle consiste à comparer n descripteurs globaux, où n est de nombre d'images de la base, avec le descripteur global de l'image requête. L'implémentation de [TAS+15] utilisée dans [SMT+18], appelée DenseVLAD, effectue cette recherche en moins de 20 ms, sans utiliser de méthode accélérée, sur la même architecture et les mêmes bases de grande taille que celles utilisés pour évaluer Active Search. De plus, le modèle se résumant à n vecteurs de taille réduite (128 pour DenseVLAD), il est particulièrement compact. En revanche, la précision de cette approche est, à images de référence identiques, généralement bien plus faible que celle des méthodes basées sur des descripteurs locaux, du fait que le pose est approximée par la pose correspondant au descripteur global le plus proche dans la base d'images. Cela est confirmé par les benchmarks présentés dans [SMT+18]. Les auteurs de ces benchmarks précisent toutefois que DenseVLAD obtient des poses dans certains scénarios où Active Search échoue (notamment en présence d'une végétation importante). Aussi préconisent-ils d'utiliser cette méthode pour estimer grossièrement la pose dans des scénarios de type navigation autonome. Il est important de souligner, toutefois, que la précision de ce type de méthode peut être améliorée en densifiant les images de la base, à l'aide par exemple de vues synthétiques [TAS+15]. Augmenter le nombre d'images de la base implique en revanche de considérer un modèle plus lourd en mémoire et d'accroître le temps de recherche du descripteur le plus proche.

Traditionnellement, les descripteurs BoW et VLAD sont utilisés pour la reconnaissance de lieux (*place recognition*). Dans ce domaine, des descripteurs générés par des CNN peuvent aussi être utilisés. Un CNN comporte en effet plusieurs couches successives d'opérations de *pooling* (`pool`) et de convolution (`conv`) et généralement, en fin de réseau, une ou plusieurs couches dites *fully connected* (`fc`). Les couches `pool` et `conv` produisent un certain nombre L d'images d'une certaine taille $W \times H$, qui peuvent être concaténées en un vecteur de taille $L \times W \times H$. Les couches `fc` produisent L valeurs qui, elles aussi, peuvent être concaténées en un vecteur de taille L. Typiquement, la couche `conv3` du réseau AlexNet [KSH12] contient 384 images de taille 13×13, donnant lieu à un vecteur de taille 64896, et la couche `fc6` contient 4096 valeurs. Intuitivement, si un réseau a été entraîné à réaliser une certaine tâche (reconnaissance d'objet, catégorisation sémantique de lieux, etc.), les vecteurs agrégeant les différentes couches de ce réseau sont, d'une certaine manière, représentatifs de l'image donnée en entrée du réseau du point de vue de la tâche apprise, et peuvent à ce titre être qualifiés de descripteurs. Étonnamment, il s'avère qu'en pratique, les CNN sont versatiles et transférables, c'est-à-dire que des descripteurs extraits d'un réseau entraîné à réaliser une certaines tâche peuvent très bien s'avérer utiles à résoudre d'autres tâches [RASC14].

Sünderhof et al. ont ainsi montré dans [SDS+15] que le descripteur correspondant à la couche `conv3` du réseau AlexNet entraîné à reconnaître des objets est particulièrement performant pour la reconnaissance de lieux basée sur la distance (cosinus des vecteurs normalisés) entre le des-

cripteur associé à l'image requête et les descripteurs associés aux images de la base. La couche `conv3` s'avère posséder un "degré d'abstraction" intéressant pour cette tâche. Le descripteur issu de cette couche est en effet particulièrement robuste aux changements d'apparence induits par l'heure de la journée, les saisons et les conditions climatiques. Une couche plus élevée telle que `fc6` est plus robuste aux changements de points de vue, mais aussi trop abstraite pour distinguer des scènes de même type. Un descripteur issu de cette couche serait plus adapté, selon les auteurs, à la catégorisation de scène. À l'inverse, les descripteurs issus des premières couches du réseau sont trop proches de l'image brute et pas suffisamment abstraits pour rapprocher des images d'un même lieu ayant une apparence différente.

Je n'ai pas connaissance de travaux utilisant un descripteur CNN pour approximer la pose par celle correspondant à l'image de la base la plus proche selon ce descripteur, comme exposé ci-dessus avec le descripteur VLAD. La reconnaissance de lieux, pour être performante, requiert une relative invariance aux changements de point de vue, que nous devons justement éviter dans un contexte de calcul de pose par recherche de l'image la plus proche, au sens des éléments observés dans les images mais aussi des points de vue depuis lesquels ils ont été observés. Un descripteur CNN sera toutefois utilisé au chapitre 3 pour reconnaître des portions d'images correspondant à des façade de bâtiments et permettre un calcul de pose.

1.2.1.3 Inférence directe par réseau de neurones convolutifs

Dans [KGC15], Kendall et al. décrivent un CNN, appelé PoseNet, qui prend en entrée une image I, et rend en sortie le quaternion $\mathbf{q}$ et la position $\mathbf{x}$ de la pose correspondant à cette image. Ce réseau est entraîné à partir d'images associées à des poses issues d'une solution SFM obtenue dans un environnement donné. Il est valide pour cet environnement uniquement, et les poses sont obtenues dans le repère du nuage de points de la solution SFM (utiliser PoseNet dans un nouvel environnement oblige donc à ré-entraîner le réseau à partir d'images et de poses acquises dans cet environnement). PoseNet s'avère relativement robuste aux changements d'illumination et au flou [KGC15]. La robustesse aux occultations n'est pas mesurée dans [KGC15], mais elle peut être renforcée en introduisant des occultations arbitraires dans les images d'apprentissage [SMD+18]. La précision de PoseNet est en revanche relativement faible, et d'autant plus faible que la vue à traiter est éloignée des vues utilisées lors de l'apprentissage (problème assez similaire à celui décrit dans la section précédente). Un autre inconvénient de cette approche est inhérent à la fonction de perte (*loss function*) minimisée au cours de l'entraînement du CNN,

$$loss(I) = ||\hat{\mathbf{x}} - \mathbf{x}||_2 + \beta \left\| \hat{\mathbf{q}} - \frac{\mathbf{q}}{||\mathbf{q}||} \right\|_2 ,$$

qui introduit un paramètre de pondération β entre l'erreur de position et l'erreur d'orientation. Le réglage de ce paramètre est particulièrement délicat, et suivant la valeur choisie, tend à privilégier la précision de la position ou celle de l'orientation (voir la figure 1.1). Pour pallier ce problème, d'autres fonctions de perte ont été envisagées, telles que l'erreur de reprojection du nuage de points obtenu par SFM [KC17] ou la distance euclidienne entre les points de ce nuage exprimés dans le repère caméra *ground truth* et dans le repère caméra estimé [LWJ+18].

Il reste que ce type d'approche pose des problèmes de précision et de mise en œuvre. Par exemple, les auteurs de [SMT+18] reconnaissent avoir évalué PoseNet, sans être parvenus à obtenir des résultats compétitifs, ce qui les a incités à retirer PoseNet de leur évaluation.

1.2.2 Méthodes exploitant des données capteurs

Nous nous intéressons à présent à des méthodes utilisant une connaissance approximative de la pose, obtenue typiquement à l'aide d'un GPS et d'un magnétomètre.

1.2.2.1 Positionnement par synthèse

Le positionnement par synthèse (*Tracking-by-synthesis*) a été proposé pour la première fois par Reitmayr et al. [RD06a]. Tandis que leur méthode repose sur un appariement de points de

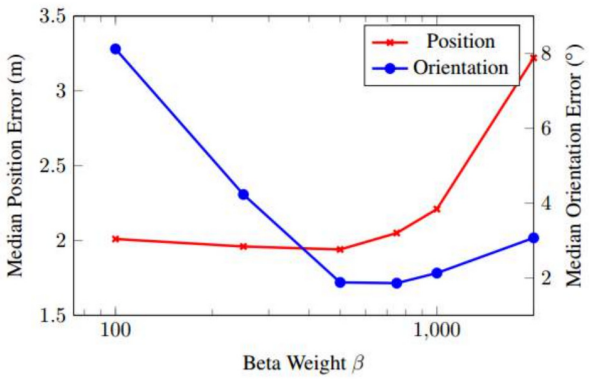


FIGURE 1.1 – Problème du réglage du paramètre β dans la méthode PoseNet [KGC15].

contours (*edgels*), nous avons tenté de prendre en compte des points de texture [19]. Le principe de cette méthode est illustré en figure 1.2. On suppose qu'un modèle polyédrique texturé de la scène est disponible (figure 1.2, en haut à gauche), ainsi qu'une estimée approximative de la pose. Une image du modèle 3D est synthétisée à l'aide d'un moteur de rendu utilisant la pose approximative (étape 1 en figure 1.2). Des points sont alors détectés dans l'image synthétique et dans l'image courante, puis appariés par corrélation croisée dans les voisinages des points, les deux images étant supposées proches (étape 2). Chaque point détecté dans l'image synthétique est associé à un unique point 3D du modèle, que l'on peut calculer par intersection des faces du modèle avec le rayon inverse issu du point (étape 3). La pose est alors calculée en utilisant les paires de points 3D-2D ainsi obtenues (étape 4).

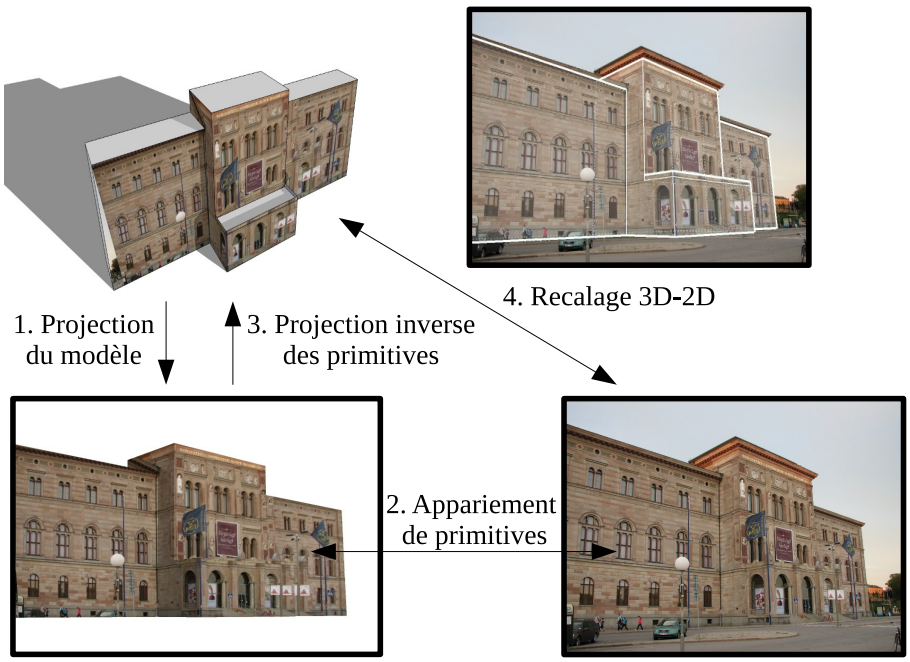
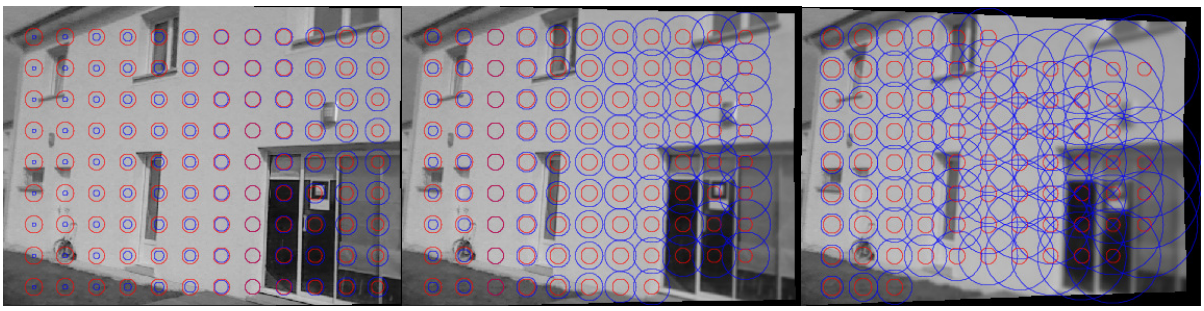


FIGURE 1.2 – Principe du positionnement par synthèse.

Le positionnement par synthèse revient donc à estimer le mouvement correctif à appliquer à la pose approximative afin que l'image rendue selon cette pose soit transformée en l'image requête. Un des intérêts de cette approche est que les faces du modèle, et donc les points supposés visibles dans l'image courante, sont données en sous-produit du rendu, tandis qu'avec les approches utilisant un nuage de points (section 1.2.1.1), tous les points du nuage 3D sont susceptibles d'être examinés lors de la phase d'appariement, ce qui est coûteux en temps de calcul et augmente le risque d'ambiguïté.

Un autre intérêt de cette approche est que l'étape d'appariement bénéficie de la proximité



$\Rightarrow 1024 \times 1024$ ($d_i = 8.6$m) $\Rightarrow 512 \times 512$ ($d_i = 17.2$m) $\Rightarrow 216 \times 216$ ($d_i = 34.5$m)

FIGURE 1.3 – Floutage avant rectification d'une texture d'un modèle 3D utilisée par un algorithme de suivi par synthèse, pour trois niveaux de la pyramide mipmap. Le diamètre des cercles rouges est proportionnel au niveau de flou déjà présent dans l'image (calculé aux centres des cercles, en utilisant la courbe de réponse du flou optique de la caméra), celui des cercles bleus au niveau de flou à appliquer à la texture selon son niveau dans la pyramide mipmap.

entre l'image synthétique et l'image requête. Cela est particulièrement intéressant lorsque des *edgels* sont utilisés, car il n'existe pas de descripteur équivalent à SIFT pour les contours. Dans [RD06a], les correspondances sont obtenues par corrélation croisée 1D le long des normales aux points de contours. Dans le cas des points, un descripteur tel que LIFT ou SIFT pourrait être utilisé mais ces descripteurs, nous l'avons vu, sont source d'ambiguïté dans l'étape d'appariement en présence de motifs répétés. Les approches présentées en section 1.2.1.1 exploitent des modèles SFM ou SLAM contenant des points en dehors du bâti. En revanche, le positionnement par synthèse n'utilise que le bâti, sur lequel les motifs répétés (typiquement les fenêtres) sont les plus présents. Un autre intérêt de pouvoir utiliser la corrélation croisée sur des points de type FAST [RD06b] ou Harris [HS88] plutôt que de calculer et apparier des descripteurs de type SIFT, est le gain obtenu en terme de temps de calculs, qui rend possible l'utilisation de cette méthode sur un dispositif léger. En revanche, la méthode est relativement coûteuse en mémoire utilisée puisque le modèle texturé complet de l'environnement doit y être stocké [6].

Les avantages liés à la proximité entre l'image synthétique et l'image vidéo sont toutefois à nuancer. Nous avons en effet montré dans [19] que la détection de points d'intérêt par la méthode de Harris [HS88] ou la méthode FAST [RD06b] est généralement moins répétable entre une image synthétique brute et une image réelle qu'entre deux images réelles également éloignées. Cela provient de la présence de flou optique dans l'image vidéo, plus ou moins prononcé suivant la distance de la caméra à la scène, qui peut être différente entre l'endroit depuis lequel on acquiert la texture du modèle et l'endroit où l'on se trouve au moment du calcul de pose. Nous avons montré qu'en appliquant à la volée un flou optique à l'image synthétique, la répétabilité de ces détecteurs augmentait considérablement (jusqu'à 100%). L'application du flou adapté à la profondeur est très rapide dans notre méthode, car elle repose sur la technique pyramidale du MIP Mapping disponible dans OpenGL (voir la figure 1.3). Malheureusement, elle nécessite de calibrer la courbe de réponse du flou optique de la caméra ce qui, bien que nous ayons proposé une méthode simple pour réaliser cette opération à l'aide d'un marqueur filmé à différentes profondeurs, représente une tâche supplémentaire à accomplir préalablement à l'utilisation du système de positionnement.

Un autre inconvénient du positionnement par synthèse est qu'il peut échouer si la précision de la pose initiale est trop faible pour retrouver les points de l'image vidéo dans l'image synthétique. L'appariement par corrélation est en effet contraint à une fenêtre de recherche (1D dans le cas des points de contours, 2D dans le cas des points de texture). Une fenêtre trop grande augmenterait le risque de confondre le correspondant recherché avec un autre point. À l'inverse, une fenêtre trop

6. Pour être plus précis, nous avons implémenté et utilisé avec succès cette méthode sur un smartphone [53], avec toutefois des modèles 3D relativement petits. La génération de l'image synthétique était très rapide, mais le temps de transfert du GPU (utilisé pour synthétiser l'image) vers le CPU (utilisé pour la traiter) constituait de manière inattendue la plus grande part du temps de traitement requis par le programme.

petite augmenterait le risque que le correspondant se trouve en dehors de la zone de recherche.

1.2.2.2 Approches *ad hoc* exploitant des labels sémantiques

Assez récemment, des auteurs ont repris le principe du positionnement par synthèse, sans toutefois explicitement générer des images de synthèse d'un modèle 3D texturé. Ces méthodes exploitent divers indices extraits de l'image (contours, couleur etc.) ainsi que des labels sémantiques associés aux pixels. Ainsi dans [APV+15], une pose initiale obtenue à l'aide d'un GPS est raffinée en alignant un modèle 2,5D grossier (empreintes au sol des bâtiments + hauteurs) avec des bâtiments visibles dans l'image requête. L'orientation de la caméra est calculée à partir des points de fuite détectés dans l'image et des hypothèses de position sont générées en appariant les crêtes verticales du modèle aux segments de droite verticaux détectés dans l'image. Une segmentation sémantique utilisant un classifieur SVM sur des descripteurs d'image locaux permet de faire la distinction entre les pixels appartenant à une façade et les pixels appartenant à l'arrière-plan. La vraisemblance de la classification au regard des faces projetées du modèle est alors maximisée sur toutes les hypothèses de pose. Dans cette approche, la précision du recalage dépend de la segmentation des pixels, qui est généralement bruitée et ne sépare pas les façades adjacentes. De plus, les éléments structurels des façades (fenêtres, portes, etc.) ne sont pas détectés par le classifieur (ou plus exactement, sont classifiés en façade, alors qu'ils sont susceptibles de contribuer à un recalage plus précis.

Chu et al. [CWUF16], en revanche, exploitent cette information structurelle pour mieux estimer la pose, ainsi que les paramètres, non connus *a priori*, d'un modèle 3D définissant la géométrie d'un bâtiment visible dans l'image (hauteur de chaque étage, position verticale des fenêtres et des portes, etc.). De même que dans [APV+15], la méthode, appelée HouseCraft, suppose que la pose est connue grossièrement grâce à des données GPS et utilise les empreintes au sol géoréférencées du bâtiment pour aider à retrouver sa géométrie. L'estimation de la pose et de la géométrie est formulée en un problème d'inférence dans un champ aléatoire de Markov, encourageant la projection du modèle 3D à être alignée avec plusieurs indices détectés dans l'image (contours, fenêtres et porte, etc.) et à contenir des couleurs différentes de celles en dehors de la projection, dans plusieurs images extraites de GoogleStreetView autour du bâtiment. Malheureusement, la complexité de l'inférence, basée sur une recherche discrétisée des paramètres optimaux, rend cette méthode inutilisable dans un contexte de localisation urbaine temps réel.

1.3 Indices de performance considérés

Les indices de performance utilisés dans l'état de l'art ci-dessus correspondent essentiellement à quatre critères :

un critère de précision (CP) tenant compte des attentes en RA (pas d'effet de tremblement etc.),

un critère de stabilité (CS) pour être stable au sens décrit plus haut, un système de positionnement visuel doit être robuste aux conditions d'éclairage (incluant l'alternance jour/nuit) et météorologiques (climat, saison, ...), à de grands changements de point de vue entre les images utilisées pour générer le modèle et l'image réquête, ainsi qu'à la présence d'objets et/ou de piétons dans l'image requête, non représentés dans les images utilisées pour générer le modèle (et vice-versa),

un critère de compacité du modèle (Cm) le modèle doit être suffisamment compact pour autoriser une implémentation du système de positionnement visuel sur une architecture légère (typiquement un smartphone),

un critère de temps de calcul (Ct) la RA requiert un traitement en temps réel des images utilisées par le système de positionnement visuel, idéalement sur un dispositif léger [7].

7. Les deux premiers critères me semblent plus fondamentaux que les deux derniers (qui sont plus ou moins critiques selon l'architecture matérielle utilisée), mais ceux-ci ne peuvent être négligés dans le cadre de la RA.

De plus, les méthodes utilisant la connaissance d'une pose approximative doivent être robuste à l'imprécision des mesures obtenues par les capteurs externes (GPS + magnétomètre). Pour ces méthodes, nous ajoutons donc la robustesse aux imprécisions de la pose initiale au critère de stabilité CS.

Méthode	**CP**	**CS**	**Cm**	**Ct**
Descripteurs locaux	✓			
Descripteur global		✓	✓	✓
Regression par CNN		✓	✓	✓
Positionnement par synthèse	✓			✓
Ad hoc [APV+15]			✓	✓
Ad hoc [CWUF16]	✓	✓	✓	

TABLE 1.1 – Satisfaction aux critères de performance visés par les différentes catégories de méthodes présentées dans l'état de l'art.

Les travaux présentés dans ce mémoire contribuent au positionnement purement visuel. Ils sont motivés par la volonté de satisfaire à l'ensemble des critères énumérés ci-dessus. En effet, si l'on reprend les différents systèmes de positionnement visuel décrits dans l'état de l'art, aucun ne satisfait pleinement aux quatre critères (Table 5.1) :

- utiliser un nuage de points SFM ou SLAM associé à des descripteurs locaux permet d'être précis quand l'étape d'appariement réussit, mais ne respecte aucun des critères CS (en raison notamment de la non invariance des descripteurs locaux aux conditions d'éclairage, à la météo et aux larges changements de point de vue), Cm et Ct,
- calculer la pose en comparant un descripteur global de l'image aux descripteurs d'une base de référence est relativement stable, rapide et repose sur un modèle compact (un descripteur et une pose par image de la base de référence), mais ne satisfait pas au critère CP. Ou alors, si on densifie à outrance la base de référence, il est peut-être possible de respecter le critère CP (bien que je ne connaisse pas d'expérimentation le prouvant) mais peut-être plus les critères Cm et/ou Ct.
- PoseNet satisfait CS, Cm et Ct, mais n'est définitivement pas précis,
- le positionnement par synthèse est rapide et potentiellement précis ; il est invariant aux changements de point de vue par nature, et la corrélation croisée utilisée pour l'appariement des points est supposée invariante aux changements d'illumination (en pratique, je ne connais pas d'étude concernant l'alternance jour/nuit et les conditions climatiques) ; toutefois l'utilisation d'une fenêtre de recherche, nous l'avons vu, rend la méthode vulnérable aux imprécisions de la pose initiale : le critère CS n'est donc pas entièrement satisfait. Cette méthode peut par ailleurs réclamer beaucoup de mémoire pour le stockage du modèle 3D texturé ;
- un positionnement *ad hoc* utilisant la sémantique tire profit de l'encodage des variations d'éclairage et des conditions météorologiques par le CNN. La précision de [APV+15] peut en revanche être insuffisante du fait de ne pas distinguer les façades adjacentes et de ne pas utiliser les structures fines des façades, tandis que [CWUF16] est très lent. Par ailleurs, [APV+15] ne satisfait pas entièrement au critère de stabilité CS, du fait que cette méthode peut échouer lorsque les données GPS utilisées sont trop imprécises. La méthode HouseCraft [CWUF16] est plus robuste aux erreurs d'imprécision du GPS, grâce au fait que des hypothèses discrètes de position de la caméra sont considérées dans un voisinage de la position donnée. Mais ce gain en précision est obtenu au prix de temps de calcul rédhibitoires (19 s par image, sur une machine dont les auteurs précisent seulement qu'elle est équipée d'un GPU).

Pour cette raison, j'utilise une lettre majuscule pour les critères CP et CS et minuscule pour les critères Cm et Ct.

1.4 Vue d'ensemble et justification de nos travaux

Bien qu'étant entièrement visuelle, notre méthode s'inscrit dans l'esprit du positionnement par synthèse et se rapproche le plus de la méthode HouseCraft [CWUF16]. Cette méthode est potentiellement précise et stable, et utilise un modèle compact de la scène (géométrie paramétrée des bâtiments incluant les portes et fenêtres). Elle a en revanche été conçue pour reconstruire un modèle 3D de bâtiment à partir de ses plans au sol, de photographies du bâtiment et de données GPS. Le temps de calcul utilisé pour produire un résultat à partir de ces données n'est absolument pas compatible avec la RA. Une grande partie des traitements est consacrée à la recherche discrétisée des paramètres optimaux du modèle et de la pose. Les paramètres du modèle peuvent être connus dans notre contexte, mais le calcul de la pose par optimisation discrète, conséquence de l'imprécision du GPS, reste problématique. Deux solutions peuvent être envisagées pour contourner cette difficulté :

— la première consisterait à estimer la pose initiale en utilisant une méthode de vision par ordinateur susceptible de produire des poses plus précises que celles pouvant être obtenues avec des capteurs, ce qui permettrait de limiter le domaine de discrétisation de la pose considéré lors de l'inférence ;

— la seconde consisterait à remplacer la méthode d'inférence utilisée dans [CWUF16] par un calcul direct (non discrétisé) de la pose robuste aux imprécisions de l'estimée initiale.

Nous avons choisi d'attaquer le problème par les deux bouts. Notre approche comporte ainsi deux étapes :

— étape 1 (chapitre 3) : une pose grossière (quoique plus précise que dans les cas défavorables du GPS) est obtenue en exploitant les balises visuelles à gros grain que constituent les façades de bâtiment ;

— étape 2 (chapitre 4) : une pose précise est calculée par recalage 3D-2D d'un modèle géométrique et sémantique des façades, selon un schéma espérance-maximisation (EM) initié par la pose approximée.

L'étape 1 utilise un descripteur CNN de niveau intermédiaire entre les descripteurs locaux et les descripteurs globaux présentés en section 1.2.1. Il est calculé sur des boites supposées englober les façades, de tailles intermédiaires entre un patch (descripteur local) et l'image (descripteur global). Détecter des objets dans une image sous forme de boites englobantes est l'une des tâches les plus anciennes réalisées par des CNN [GDDM14, Gir15, RHGS15]. La plupart des architectures proposées intègrent un module de proposition de boites susceptibles de contenir "un objet" (*object proposal*), sans pouvoir en dire plus sur la classe de cet objet (le CNN s'en charge par la suite). Des primitives de bas niveau tels que des superpixels (Selective Search [UvdSGS13]) ou les contours de l'image (Edge Box [ZD14]) sont utilisées à cette fin.

La proposition d'objet a par exemple été exploitée dans [SSJ+15] pour améliorer les performances de la reconnaissance de lieux. La méthode Edge Box est utilisée pour générer des boites de proposition d'objet, aussi bien dans les images de référence que dans l'image requête. La portion d'image contenue dans chacune des boites proposées est redimensionnée de manière à pouvoir être donnée en entrée du réseau AlexNet. Enfin, tout comme pour l'obtention du descripteur global présenté plus haut [SDS+15], la couche `conv3` est utilisée pour générer un descripteur associé à chaque boite. Les descripteurs obtenus dans chaque image de la base de référence sont appariés aux descripteurs obtenus dans l'image requête par recherche du plus proche voisin selon le cosinus de l'angle entre les descripteurs et l'application de la contrainte de choix mutuel. La similarité de l'image requête avec une image de la base est mesurée à l'aune des scores d'appariement (cosinus) obtenus pour les paires de descripteurs retenus pour ces images, ainsi que d'un critère de similarité de forme calculé sur chaque paire de boites appariées. Le temps de calcul consacré à la recherche de l'image la plus proche est donc plus élevé que dans [SDS+15], mais reste très inférieur à celui d'un appariement de descripteurs locaux, du fait que le nombre de boites utilisées (par exemple, entre 50 et 100 dans [SSJ+15]) est généralement bien plus faible que le nombre de points considérés lors d'un tel appariement (généralement plusieurs milliers). La robustesse aux changements de point de vue, en revanche, est accrue par apport à [SDS+15], en raison du fait qu'un changement de perspective impacte moins l'apparence des objets isolés

que leurs positions relatives dans l'image, c'est-à-dire l'apparence globale de l'image.

Cette approche étant plus robuste aux changements de point de vue que la méthode présentée dans [SDS+15], elle est d'autant moins adaptée au calcul de pose par proximité avec une image de référence dont la pose est connue. Les boites englobant les façades ne peuvent pas non plus être utilisées en l'état pour un calcul de pose par résolution du problème PnP puisque, à la différence des descripteurs locaux dont les centres des patchs sont considérés comme les correspondants dans l'image des points identifiés dans le modèle 3D, les centres des boites englobantes ne correspondent généralement pas aux centres des rectangles représentant les façades dans le modèle, du fait de la déformation perspective. En revanche, si l'image est rectifiée de sorte les façades apparaissent en vue fronto-parallèle, alors les boites englobantes de l'image rectifiée peuvent effectivement correspondre à des projections centrales des rectangles du modèle. Le centre des boites 2D, mais aussi leurs quatre coins, sont alors sensés correspondre aux centres et aux coins des rectangles 3D, ce qui permet un calcul de pose par résolution du problème P4P à partir d'une seule façade correctement détectée.

Ainsi, dans le même esprit que l'*object proposal*, nous proposons une méthode de *façade proposal*, conçue pour générer des propositions de boites autour des façades visibles dans une image rectifiée. Cette méthode s'appuie sur des indices géométriques, photométriques et sémantiques de l'image, propres aux façades et pouvant être évalués rapidement. Un réseau de neurones spécifique est ensuite appliqué à chaque boite proposée, pour décider plus finement si elle contient effectivement une façade, et dans le cas positif, un descripteur CNN de la portion d'image contenue dans la boite est calculé. En pratique, les bords des boites détectées peuvent ne pas correspondre exactement aux bords de la façade, ne serait-ce que parce que ces bords ne sont pas toujours visibles dans l'image requête (ils peuvent être situés en dehors de l'image ou occultés par d'autres objets de la scène). Une calcul P4P utilisant les coins de la boite peut donc être imprécis, mais la pose obtenue présente généralement l'intérêt de projeter le modèle "pas trop loin" de sa position dans l'image, condition favorable à la réalisation de l'étape 2. À l'inverse, certaines mesures capteurs peuvent mener à une situation où le modèle est projeté loin de la façade visible dans l'image, parfois même en dehors de l'image.

Rectifier une image relativement à deux directions orthogonales de la scène se ramène à détecter les points de fuite de l'image correspondant à ces deux directions. Aussi proposons-nous une méthode rapide et efficiente pour calculer le zénith et les points de fuite horizontaux d'une image monoculaire non calibrée (chapitre 2). Une analyse statistique fine, issue de la théorie du Gestalt [DMM07], de la distribution des segments de droite détectés dans l'image nous permet d'obtenir ces points avec une précision supérieure, en particulier lorsque des objets fabriqués par l'homme sont présents à hauteur de vue, à la précision des méthodes de l'état de l'art les plus performantes jusqu'ici.

Notre méthode de proposition de façade est robuste aux conditions d'éclairage et météorologiques parce qu'elle repose sur des indices, en particulier sémantiques, robuste à ces conditions. Elle est par nature invariante aux changements de points de vue du fait qu'elle opère sur des images rectifiées. La reconnaissance des façades détectées bénéficie également d'une invariance aux conditions d'éclairage et météorologiques et aux changements de point de vue, ces différents facteurs étant encodés par le réseau neuronal convolutif utilisé pour calculer les descripteurs. L'étape 1 est donc stable au sens défini plus haut. Le modèle utilisé est par ailleurs très léger puisque constitué d'une liste de descripteurs (un descripteur par façade de référence) de taille 2048. Enfin, le nombre de façades visibles dans une image étant généralement faible (typiquement moins de 10), l'étape d'appariement des descripteurs est très rapide (en moyenne 50 ms sur un processeur I7-3520M associé à une carte graphique Nvidia TITAN X). La proposition de façade est, en revanche, un peu plus lente (près d'une demi-seconde sur la même architecture), mais reste compatible avec la RA (nous y reviendrons).

L'étape 2 bénéficie de toute la précision qu'un recalage basé modèle peut apporter. La carte sémantique sur laquelle repose cette étape est générée par un CNN de type auto-encodeur, également invariant aux conditions d'éclairage et météorologiques. Le fait d'opérer dans un cadre bayésien permet à la fois de n'ajouter qu'un très faible poids au modèle (quelques paramètres de

modèles de mixtures de gaussiennes généralisées – GGMM) et d'être robuste aux initialisations imprécises de la pose (une gaussienne généralisée possède un support infini). Nous avons par ailleurs pu obtenir une excellente robustesse aux occultations, en permettant aux poids des GGMM de varier au cours des itérations de l'EM, tout en restant attachés à une distribution de Dirichlet. L'étape 2 est donc également stable, et sa durée moyenne sur l'architecture décrite précédemment est d'environ 100 ms.

La durée cumulée des étapes 1 et 2, à laquelle il faut ajouter le temps consacré à la détection des points de fuite, est de l'ordre de la seconde sur un PC relativement performant en 2017. Cela est très inférieur aux 19 s indiquées par les auteurs de HouseCraft [APV+15] (bien qu'il soit difficile de comparer les deux durées, les auteurs de HouseCraft ne précisant pas l'architecture utilisée), mais reste trop élevé au regard des exigences la RA qui réclame un temps de traitement inférieur à 100 ms (10Hz, idéalement 30 Hz, la cadence vidéo). Toutefois, une fois la pose initialisée (étape 1) et affinée (étape 2) dans l'image 1, l'étape 2 uniquement peut être appliquée à l'image 2, en initialisant l'EM avec la pose précise obtenue pour l'image 1, *a priori* très proche de la pose correspondant à l'image 2. De même, seule l'étape 2 est nécessaire pour calculer la pose dans l'image 3 à partir de la pose obtenue dans l'image 2. Ce procédé peut se poursuivre indéfiniment jusqu'à ce que l'étape 2 échoue (ce qui peut se mesurer de diverses manières, par exemple en mesurant l'IoU – *Intersection over Union* [ZD14] entre la sémantique extraite de l'image et la sémantique projetée). Dans ce cas, il suffit de rappeler la procédure utilisée à l'étape 1 pour réinitialiser la pose et poursuivre avec l'étape 2 uniquement. Cette stratégie revient à utiliser 1 s pour initialiser le processus et le réinitialiser ponctuellement, puis 100 ms par image entre les réinitialisations. Je qualifierais un tel scénario de "compatible avec la RA" dans la mesure où il me semble que la plupart des utilisateurs accepteraient d'attendre une seconde pour démarrer une application. Certains GPS pour véhicules mettent parfois plusieurs secondes, voire plusieurs minutes, à recevoir un signal valide, ce qui n'empêche pas que ces appareils soient utilisés en masse. Par ailleurs, l'étape 2 étant stable, il est probable que les réinitialisation soient rares, tant que des façades de référence sont visibles dans l'image vidéo.

L'implémentation de ces méthodes sur un smartphone nécessiterait toutefois certaines adaptations. Si certains appareils mobiles sont aujourd'hui dotés d'unités de calcul dédiées au traitement d'image (puces Pixel Visual Core dans le smartphone de Google, calculateur Nvidia Drive PX pour la conduite autonome), et que des travaux sont en cours pour adapter des CNN aux téléphones portables [TCP+18], il est probable que l'étape 1 ne puisse guère être utilisée immédiatement sur un smartphone ou une tablette en un temps compatible avec la RA. On peut en revanche envisager d'utiliser l'étape 2, couplée à un GPS et des magnétomètres, pour les étapes d'initialisation et de réinitialisation. 100 ms sur un PC se transformeraient probablement en quelques secondes sur un smartphone puissant, mais cette durée reste compatible avec la RA pour la phase d'initialisation. Le système deviendrait lié à la disponibilité de données GPS mais pourrait être initialisé et réinitialisé dans des conditions favorables (ciel dégagé, signal GPS peu obstrué, ...). En revanche, quelques secondes par image est beaucoup trop long pour utiliser l'étape 2 en temps réel suite à l'étape d'initialisation.

Il faut remarquer à ce stade que l'étape 2 est inutilement lourde pour actualiser la pose dans les images vidéo, une fois qu'une initialisation *précise* a été obtenue. En effet, de nombreux traitements opérés au cours de cette étape visent à ce qu'elle soit robuste aux changements d'éclairage et de météo entre la phase d'acquisition du modèle et le moment de son utilisation, et également robuste à une initialisation de la pose éloignée de la pose courante. Or, deux images consécutives d'une vidéo ne présentent généralement qu'une infime variation en termes d'éclairage, de météo et de pose. Dans ces conditions, n'importe quelle méthode d'odométrie visuelle (suivi de plans [32], SLAM visuel [DM02], ...) est à même d'actualiser la pose. L'odométrie visuelle (du grec *hodos* – voyage et *metron* – mesure) permet d'estimer le *mouvement* de la caméra entre deux images. Elle est donc tributaire d'une initialisation *précise* de la pose et peut souffrir d'un phénomène de dérive nécessitant des réinitialisations régulières, mais l'étape 2 permet justement de remédier à cela. En revanche, elle est très rapide et plusieurs méthodes d'odométrie visuelle sont disponibles depuis des années sur téléphone portable : le suivi de plan peut être vu comme

une variante plus rapide du suivi par synthèse (l'image synthétique est remplacée par l'image précédente de la vidéo, ce qui évite une grande partie des traitements mais ne permet pas de corriger une pose imprécise) que nous avons nous-mêmes implémenté sur smartphone [53], et il existe par ailleurs plusieurs systèmes de SLAM visuel spécialement conçus pour être exécutés en temps réel sur des téléphones portables [KM09, VARS14] (dans notre contexte, seules les poses obtenues par ces systèmes sont utiles, le nuage de point généré en sous-produit ne nous intéresse pas, bien que l'on ne puisse se passer de son calcul dans le cadre d'un SLAM visuel).

Nous avons été les premiers, en 2000, à proposer une méthode d'odométrie visuelle temps réel n'utilisant ni capteurs, ni marqueurs. Notre méthode, basée sur le suivi automatique d'un plan texturé de la scène [32], a d'ailleurs été couronnée par un *Lasting Impact Award* à ISMAR'2013 (International Symposium on Mixed and Augmented Reality). Cette méthode étant aujourd'hui bien ancrée, je ne la mentionnerai que brièvement en introduction du dernier chapitre du mémoire (chapitre 5). En revanche, je présenterai plus en détails dans ce chapitre une procédure moins connue, concomitante à cette méthode, qui me semble encore aujourd'hui sous-exploitée : il s'agit d'une procédure de sélection de modèle de mouvement, permettant de décider si la caméra a subi un mouvement stationnaire, rotationnel ou général entre deux images. Cela implique de ne pas estimer plus de paramètres de mouvement que nécessaire, défaut inhérent à la plupart des méthodes de positionnement anciennes ou actuelles, visuelles ou basées capteurs. Ne pas utiliser le modèle de mouvement optimal se traduit dans la plupart des cas par un effet de tremblement plus ou moins prononcé de la scène virtuelle par rapport à la scène réelle, par exemple lorsque la caméra est fixe et que le modèle général est considéré.

La connaissance du modèle de mouvement a par ailleurs une portée plus large : elle permet de détecter les cas singuliers d'une méthode de reconstruction multioculaire (de type SFM ou SLAM, toutes deux basées sur la parallaxe) ou monoculaire (considérant un mouvement stationnaire ou purement rotationnel de la caméra). Nous terminons ainsi le chapitre 5 en présentant une méthode de SLAM visuel semi-interactive, reposant sur les structures planes de la scène et nécessitant de distinguer les mouvements stationnaires ou rotationnels de la caméra (phases de *mapping* monoculaire) des mouvements généraux (phases de *localization*).

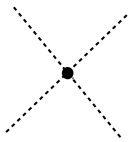
Nous avons dit au début de cette introduction générale que le SLAM visuel est "rarement directement utile à la RA". Nous avons employé le terme *rarement*, car il existe quelques applications de RA pour lesquelles il peut être utile de positionner des objets virtuels à la volée, au sein d'une carte reconstruite en temps réel : on peut penser, par exemple, à un paysagiste ou un architecte d'intérieur qui se rend tous les jours chez des clients différents et réalise sur place des esquisses de projets d'aménagement, visualisés en temps réel (par le professionnel et/ou par les clients potentiels) à l'aide d'un périphérique quelconque de RA. Des plantes, des meubles etc. virtuels peuvent dans ce contexte être placés en temps réel au sein de la carte en cours de construction (nuage de points ou, dans notre cas, ensemble de surfaces) et visualisés en temps réel en même temps que la carte est utilisée pour le calcul de pose. Nul besoin, dans cet exemple, de relier la pose obtenue à un repère global pré-existant.

Nous avons de plus utilisé le terme *directement*, car nous avons vu que les nuages de points obtenus par SLAM sont indirectement, au même titre que les nuages de points obtenus par SFM, utiles aux méthodes de positionnement reposant sur ce type de modèle (approches présentées en section 1.2.1.1). De même, notre méthode de SLAM visuel semi-interactive peut être utilisée pour obtenir *in situ* des modèles polyédriques texturés utiles au positionnement. De tels modèles peuvent en effet être employés immédiatement par notre méthode d'odométrie visuelle. Leur texture et leur géométrie peuvent également servir à générer les descripteurs CNN présentés au chapitre 3 et à calculer la pose à partir des coins des façades. Des techniques de vision par ordinateur basées par exemple sur des grammaires de façade (voir l'introduction du chapitre 4) peuvent enfin être utilisées pour extraire de ces modèles les éléments de sémantique utiles à la méthode de recalage fin présentée au chapitre 4.

La progression du mémoire semble assez naturelle dans la mesure où nous pourrions envisager un système de positionnement enchaînant les procédures décrites dans chaque chapitre, dans l'ordre d'apparition des chapitres (calcul des points de fuite, positionnement grossier, position-

nement précis, odométrie). Il convient toutefois de préciser que ces travaux (i) ont émergé dans un ordre différent de celui des chapitres, qui peuvent donc tout aussi bien être lus de manière indépendante et (ii) impactent d'autres domaines que celui de la RA : l'ensemble du mémoire concerne également la robotique et la navigation autonome ; la sélection de modèle (chapitre 5) s'applique également à la postproduction (SFM) ; la détection de points de fuite (chapitre 2) et la modélisation *in situ* (chapitre 5) présentent également un grand intérêt pour l'infographie, etc. Par conséquent, si nous avons certes tenu à être régulièrement présents à ISMAR [8], la plus prestigieuse des conférences de RA [36, 35, 32, 30, 29, 27, 25, 24, 22, 19, 18, 17, 14], une partie de nos travaux a également été diffusée dans des conférences et revues fédérant des chercheurs en vision par ordinateur (ICCV [9] [37], ECCV [10] [33, 12], ACCV [11] [38], BMVC [12] [31], VVG [13] [28]), en traitement du signal et reconnaissance des formes (ICPR [14] [26, 23, 21], ICIP [15] [13], RFIA [16] [47, 46, 44, 43, 42, 41, 40], TS [17] [7]) et en infographie (EG [18] [34, 20, 15], MVA [19] [8], TVC [20] [4], CGA [21] [6], CAVW [22] [5]).

8. International Symposium on Mixed and Augmented Reality
9. International Conference on Computer Vision
10. European Conference on Computer Vision
11. Asian Conference on Computer Vision
12. British Machine Vision Conference
13. Vision, Video and Graphics
14. International Conference on Pattern Recognition
15. International Conference on Image Processing
16. Reconnaissance des Formes et Intelligence Artificielle
17. Traitement du Signal
18. Eurographics
19. Machine Vision and Applications
20. The Visual Computer
21. Computer Graphics and Applications
22. Computer Animation and Virtual Worlds



Je ne suis que le fruit peut-être
De deux lignes qui se rencontrent.

Je n'ai rien.

On dit : partir du point,
Y arriver.

Je n'en sais rien.

Mais qui
M'effacera ?

Chapitre 2

Détection des points de fuite

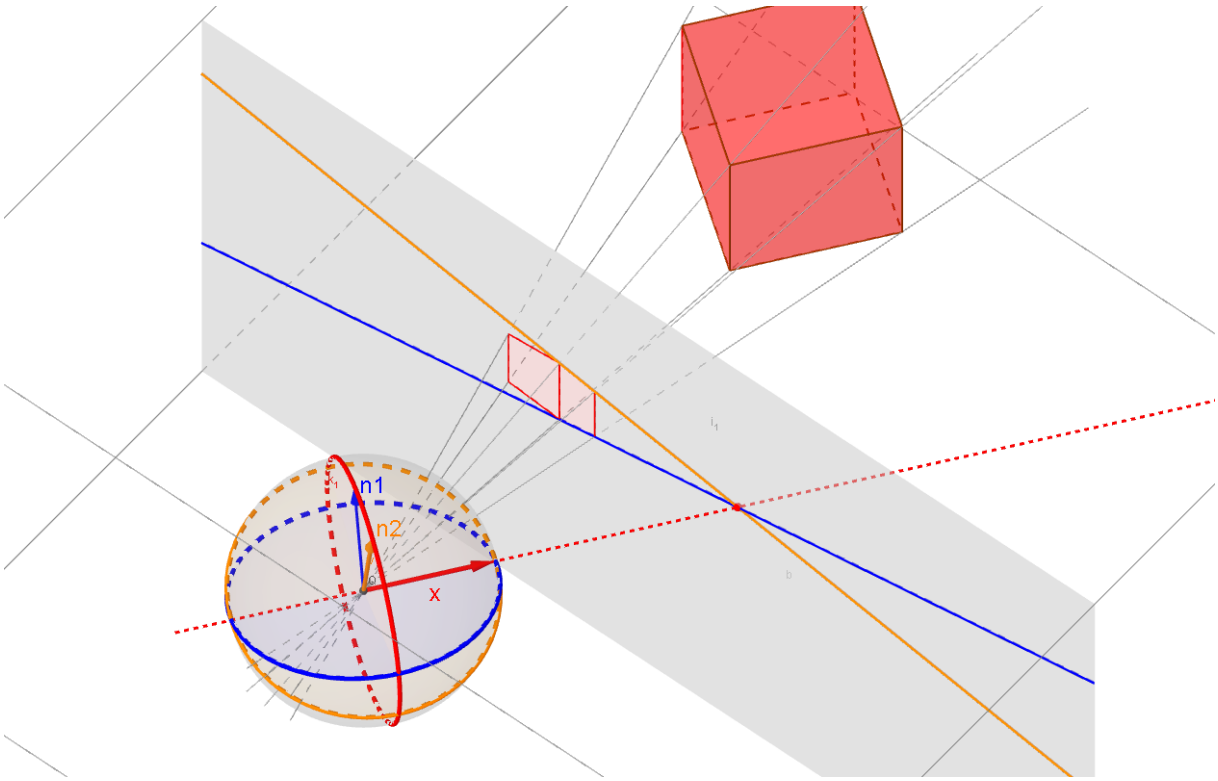


FIGURE 2.1 – Géométrie d'un point de fuite.

2.1 Introduction

Un point de fuite est un point abstrait de l'image vers lequel convergent les projections 2-D de droites parallèles dans l'espace 3-D (figure 2.1). En théorie du Gestalt [DMM07], un tel arrangement spatial d'objets perçus est appelé une *loi de groupement*, ou encore, un *gestalt*. Plus précisément, comme un segment 2-D est en soi un gestalt (en tant que groupement de points alignés), un point de fuite est qualifié de *gestalt de second ordre* [DMM07].

Détecter les points de fuite d'une image est un pré-requis de nombreux problèmes de vision par ordinateur tels que l'autocalibration [WH12], la reconstruction monoculaire [LHK09], l'odométrie visuelle [KZ02] et la navigation robotique [LSX+13], pour n'en citer que quelques-uns. À partir des points de fuite dits de Manhattan [1], il est possible de calculer la distance focale de la caméra, ainsi que son orientation par rapport au repère de Manhattan. Dans les travaux présentés aux chapitre 3 et 4, nous utilisons le zénith et les points de fuite horizontaux pour rectifier les images de telle sorte que les façades visibles dans l'image apparaissent en vue fronto-parallèle. Dans les travaux présentés au chapitre 5, les points de fuite de Manhattan sont utilisés comme support à la reconstruction *in situ* de modèles 3D du bâti. Pour toutes ces raisons, nous avons consacré beaucoup d'énergie à tenter d'améliorer la précision et la rapidité des méthodes de détection de points de fuite existantes.

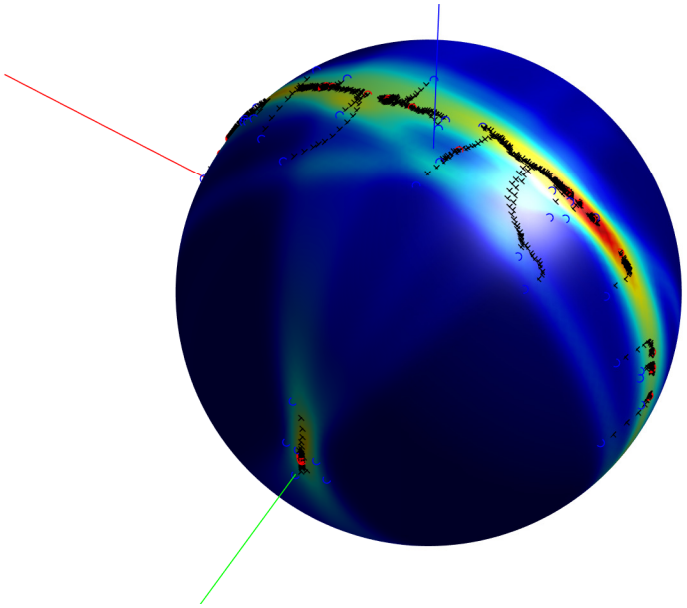


FIGURE 2.2 – Chemins parcourus sur la sphère gaussienne pendant le déroulement de l'algorithme décrit dans [48], sur une image de l'Hôtel de Ville de Nancy : les cercles bleus montrent les graines de la méthode *mean shift*, les croix noires les étapes de convergence et les cercles rouges les points de convergence, candidats aux points de fuite de Manhattan.

Une première méthode a fait l'objet de la première partie d'un article consacré à la détection de façades, présenté au "Workshop on Urban Augmented Reality" d'ISMAR 2015 [48]. Dans cette méthode, une approche bayésienne est utilisée pour rechercher les points de fuite de Manhattan sur la sphère gaussienne, en utilisant les segments de droite L détectés dans l'image. Cette recherche est découpée en deux étapes : des candidats V aux points de fuites de Manhattan sont d'abord sélectionnés sur la sphère gaussienne, en recherchant les maxima locaux de la

1. Points de fuite correspondant à trois directions de l'espace 3-D–dont la direction du zénith, orthogonales deux à deux et alignées avec les bâtiments de la scène lorsque ceux-ci peuvent être représentés par des parallélépipèdes rectangles parallèles les uns aux autres. Les scènes urbaines présentant une telle configuration sont appelées *mondes de Mahanttan* dans la littérature.

vraisemblance $p(L|V)$ par une technique de type *mean shift* (figure 2.2), avec :

$$p(L|V) = \frac{1}{C} \sum_i \exp\left(-\frac{(\mathbf{n}_i^T \mathbf{x})^2}{2\sigma^2}\right), \tag{2.1}$$

où C est un terme de normalisation, σ une constante, $\mathbf{n}_i$ la normale au plan passant par le $i^{\text{ème}}$ segment de L et le centre optique de la caméra, et $\mathbf{x}$ la direction du point de fuite V (figure 2.1). La deuxième étape consiste à rechercher le maximum *a posteriori* exprimé ci-dessous, en considérant l'ensemble discret $\mathcal{V}$ des candidats obtenus lors de la première étape :

$$\max_{(X,Y,Z)\in\mathcal{V}^3} p(L|X,Y,Z)p(Z|X,Y)p(X|Y)p(Y), \tag{2.2}$$

où $p(Z|X,Y), p(X|Y), p(Y)$ sont des priors sur les directions de Manhattan (resp. une distribution de von-Mises-Fisher, une distribution de Watson et une distribution de Kent, voir la figure 2.3), modélisés à partir d'observations obtenues sur 648 vérités terrains.

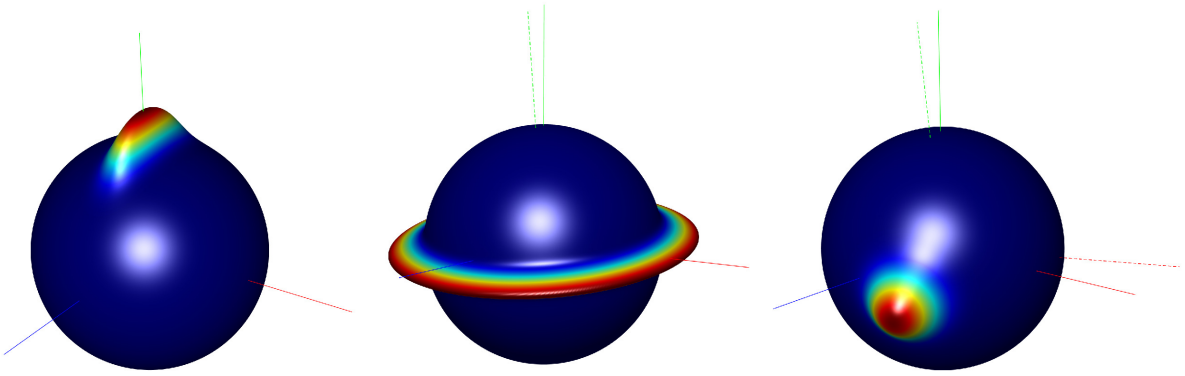


FIGURE 2.3 – Priors sur les directions de Manhattan dans le repère caméra. De gauche à droite : $p(Y)$, $p(X|Y)$ et $p(Z|X,Y)$.

Dans la deuxième partie de l'article, que nous ne détaillerons pas ici, un algorithme de détection de façades différent de celui présenté au chapitre 3 était présenté. Dans cette méthode, des coins de Harris sont détectés dans les images rectifiées (à partir des points de fuite de Manhattan obtenus) et une machine à vecteurs de support (SVM) est utilisée pour identifier les coins à angle droit parmi les coins détectés. Ces coins sont alors regroupés en régions de façades par une technique de *min-cut* rectangulaire (figure 2.4(gauche)).
La technique proposée pour l'estimation des points de fuite de Manhattan n'était pas entièrement satisfaisante car relativement lente, adossée à la connaissance des paramètres intrinsèques de la caméra et limitée à la détection des points de fuite de Manhattan. Les performances de cette méthode n'étaient en outre pas significativement meilleures que celles de la méthode la plus précise du moment [LGvGRM14]. La méthode de détection de façades ne nous satisfaisait pas totalement non plus, en premier lieu parce qu'elle avait tendance à fragmenter les façades en sous-parties, comme le montre l'image de gauche en figure 2.4, puis en raison d'un autre phénomène visible aussi dans cette image : l'idée de détecter les coins à angles droits dans l'image rectifiée était que de tels coins sont sensés apparaître uniquement sur les façades rectifiées. Or des coins à angle droit sont aussi détectés sur la façade de droite, perpendiculaire à celle de gauche et donc non rectifiée par la transformation homographique utilisée pour générer cette image. Le point intéressant est que ces coins sont regroupés autour de la ligne d'horizon (lieu des points de fuite horizontaux). La raison de ce phénomène est que les segments horizontaux dans le repère monde, situés à hauteur du centre optique de la caméra, se projettent sur la ligne d'horizon, parallèlement à celle-ci, *quelque soit leur direction dans le plan horizontal du repère monde*. Cette propriété est illustrée en figure 2.4(droite). Nous n'en présentons pas la démonstration mais celle-ci est relativement directe. Sa redécouverte "par accident" est à l'origine de l'élaboration de la

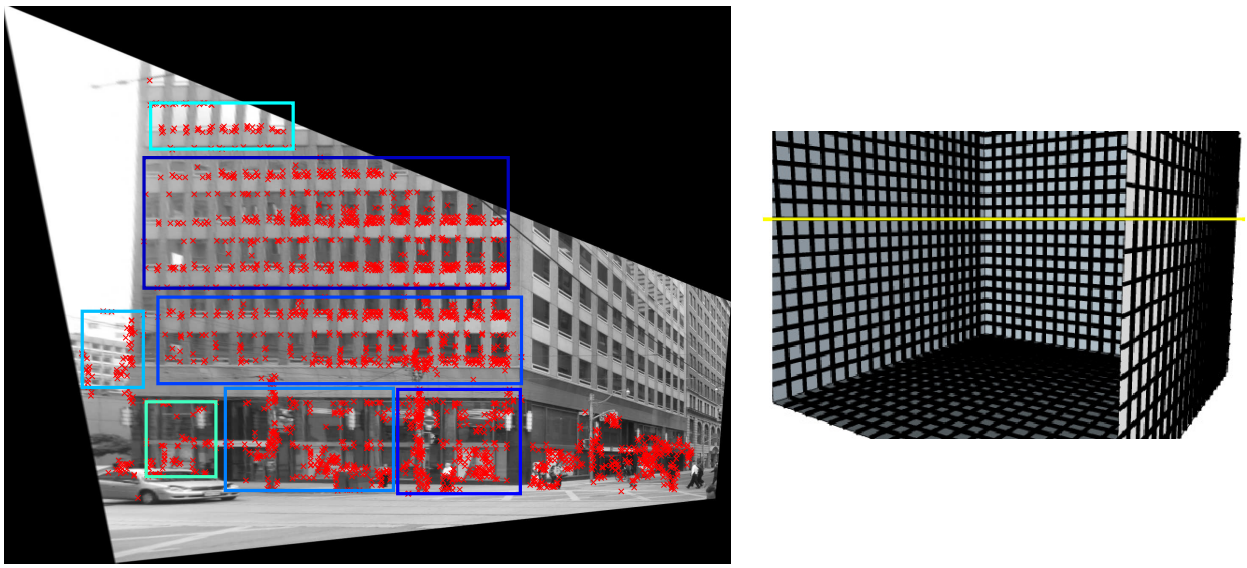


FIGURE 2.4 – À gauche : détection des coins à angle droit et regroupement en régions de façade selon la méthode présentée dans [48]. À droite : explication du fait que, dans l'image de gauche, des coins à angle droit sont aussi détectés sur la façade de droite, non rectifiée. Les segments de droite horizontaux à hauteur du centre optique de la caméra sont en effet alignés avec la ligne d'horizon, quelque soit leur direction dans le plan horizontal.

méthode de détection de points de fuite présentée dans ce chapitre. Cette dernière repose en effet sur la détection première de la ligne d'horizon en tant qu'alignement, non pas de coins, mais de segments de droites horizontaux, ou plus exactement perpendiculaires à la ligne zénithale [2], c'est-à-dire en tant que gestalt de second ordre, au même titre que les points de fuite.
Être capable de détecter la ligne d'horizon indépendamment des points de fuite est potentiellement très intéressant. Cela permet en effet de contraindre la détection des points de fuite horizontaux (généralement les seuls utiles, en plus du zénith) sur cette ligne, et de réduire ainsi le risque de fausses détections. Des auteurs avaient au préalable montré que prendre en compte la ligne d'horizon dans le calcul des points de fuite améliorait la précision par rapport aux méthodes antérieures [TBKL12]. Cependant, la ligne d'horizon étant alors détectée comme un alignement de points de fuite, c'est-à-dire un gestalt de troisième ordre, les auteurs étaient confrontés au problème de "l'œuf et la poule", qu'ils n'ont pu résoudre qu'au prix d'une optimisation d'énergie globale, coûteuse en temps de calculs.
Une première ébauche de notre méthode, publiée à Eurographics 2016 et décrite brièvement dans l'état de l'art ci-dessous, détectait la ligne d'horizon comme un pic de l'histogramme des coordonnées des segments horizontaux le long de l'axe zénithal (figure 2.5). Cette approche, bien que très rapide et facile à implémenter, était cependant confrontée à des problèmes de seuillages évidents et n'atteignait que des performances moyennes par rapport à l'état de l'art. Reconsidérer le problème par le prisme de la théorie du Gestalt a en revanche permis d'obtenir une bien meilleure précision, en fait la meilleure à ce jour sur les deux jeux de tests habituellement utilisés pour comparer les algorithmes de détection de points de fuite. Cette méthode est par ailleurs tout aussi facile à implémenter que celle présentée à Eurographics, et encore plus rapide. Elle a fait l'objet d'une publication à ECCV 2018 [12] et un article étendu est en cours de soumission à TPAMI. Notre code Matlab, permettant de reproduire les benchmarks et de détecter des points de fuite dans n'importe quelle image d'environnement fabriqué par l'homme, a en outre été rendu public [52].
Il est notable qu'en même temps que fut publié notre article d'Eurographics, d'autres auteurs ont montré qu'il est possible de détecter la ligne d'horizon *a priori* ("*Horizon-first vanishing point detection*") [ZWJ16]. Leur méthode, très différente de la notre (la ligne d'horizon est régressée

2. Droite passant par le zénith et le point principal de l'image. Une propriété bien connue en géométrie projective est que la ligne d'horizon est perpendiculaire à la ligne zénithale.

par un réseau de neurones convolutifs – voir l'état de l'art ci-dessous), montrait d'excellentes performances et de nombreuses améliorations ont dû être apportée à notre méthode, en plus du passage au cadre *a contrario*, pour parvenir à dépasser ses performances.

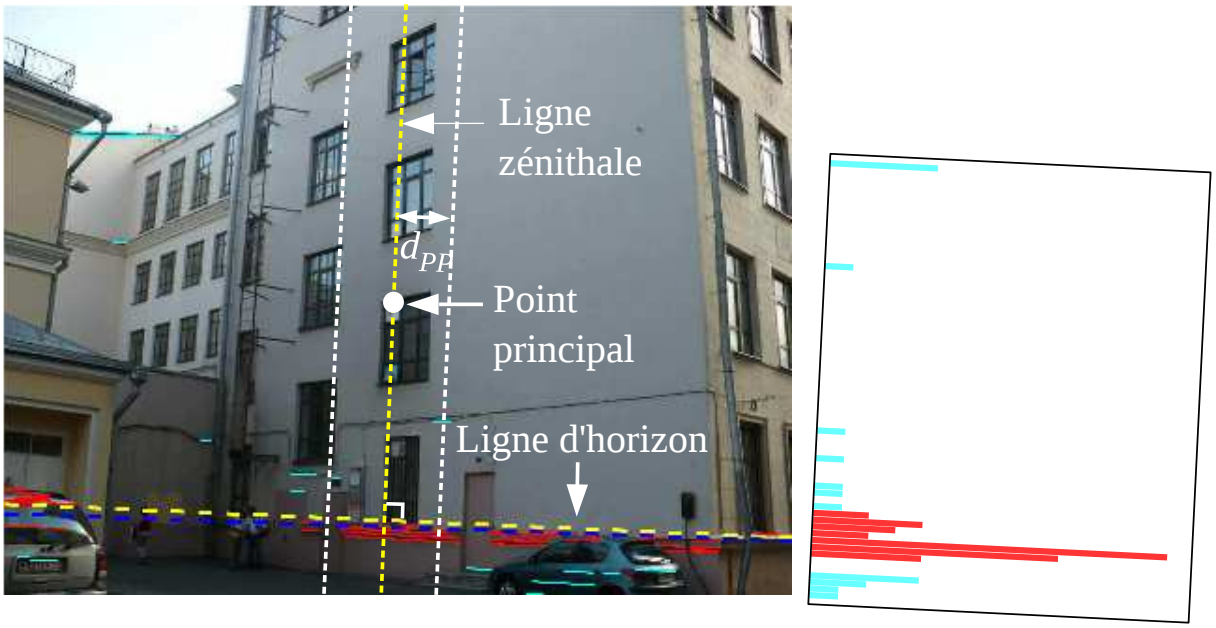


FIGURE 2.5 – Détection de la ligne zénithale et de la ligne d'horizon en tant que gestalts du second ordre : gestalt de parallélisme à l'intérieur d'une bande de demi-largeur d_{PP} pour la ligne zénithale, gestalt d'alignement perpendiculairement à la ligne zénithale pour la ligne d'horizon. L'histogramme affiché concerne la détection de la ligne d'horizon (MSM en rouge) par la méthode *a contrario*.

2.2 État de l'art et contributions

2.2.1 Approches classiques

Une vaste littérature est consacrée au problème de la détection de points de fuite dans une image non calibrée. Les auteurs de [KZ02] utilisent l'algorithme espérance-maximisation (EM) pour estimer itérativement les coordonnées des points de fuite en même temps que les probabilités des segments de droite de contribuer à chacun d'entre eux. Cependant, la méthode EM étant sensible à l'initialisation, la méthode grossière utilisée dans cet article pour résoudre cette étape est fréquemment à l'origine de résultats décevants. Plusieurs tentatives ont été faites pour obtenir une meilleure initialisation de l'algorithme EM. La méthode présentée dans [Tar09] estime des hypothèses de points de fuite à l'aide de paires de segments et calcule un ensemble de consensus en utilisant l'algorithme J-linkage. Ce principe est repris par les auteurs de [XOH13], qui introduisent une mesure probabiliste de consistance entre les segments et les points de fuite, plus performante que la mesure géométrique utilisée dans [Tar09]. Wildenauer et al. introduisent une méthode utilisant RANSAC [FB81] pour estimer trois points de fuite orthogonaux et la focale de la caméra à partir de tirages aléatoires d'ensembles de quatre segments de droite [WH12].
Toutes ces méthodes ont été comparées sur les mêmes jeux de données, York Urban (YU) [DEE08] et Eurasian Cities (EC) [TBKL12] (voir la section 2.5) et en utilisant le même protocole d'évaluation [TBKL12]. Il est délicat d'établir une vérité terrain des points de fuite, car cette tâche présente une certaine subjectivité. Pour cette raison, l'évaluation proposée dans [TBKL12] est basée sur la précision de la ligne d'horizon, calculée *a posteriori* à partir des points de fuite détectés. Il est aisé de démontrer que cette droite est perpendiculaire à la ligne zénithale (figure 2.5). La ligne d'horizon peut donc être trouvée en effectuant une recherche à une dimension (1-D) le long de la ligne zénithale, sa position correspondant à un minimum au sens des moindres carrés

pondérés, où le poids de chaque point de fuite est égal au nombre de segments contribuant à ce point de fuite. L'*erreur d'horizon* est alors définie comme la distance euclidienne maximale entre la ligne d'horizon estimée et la vérité terrain à l'intérieur des frontières de l'image, divisée par la hauteur de l'image. Pour représenter sous un même graphique les erreurs d'horizon relatives à toutes les images d'un jeu de données, un histogramme cumulé de ces erreurs est utilisé. Vedaldi et al. ont par ailleurs proposé de mesurer une valeur de précision unique pour chaque jeu de données, qui est le pourcentage de l'aire sous l'histogramme cumulé (AUC – Area Under the Curve) dans le rectangle $[0, 0.25] \times [0, 1]$ [VZ12]. Les histogrammes cumulés, ainsi que les AUC obtenues pour chaque jeu de données et pour chaque méthode sont reportés en figure 2.12. Nous pouvons constater que chaque nouvelle méthode obtient des AUC supérieures à celles des méthodes précédentes, allant de 74.34% pour YU et 68.62% pour EC avec la méthode la plus ancienne [KZ02] jusqu'à 93.45% pour YU et 89.15% pour EC avec la méthode la plus performante en 2013 [XOH13].

2.2.2 Méthodes *a contrario*

D'autres auteurs ont proposé de détecter des points de fuite significatifs au sens de la théorie du Gestlat. Cette théorie, issue de travaux en psychologie initiés au début du XXe siècle, a été formalisée mathématiquement par Desolneux et al. en 2007 [DMM07]. Selon le principe de Helmholtz, qui stipule que "nous percevons immédiatement tout ce qui ne pourrait pas se produire par hasard", une variable universelle adaptée à de nombreux problèmes de détection, le Nombre de Fausses Alarmes (NFA) a ainsi été définie. Le NFA d'un événement est le nombre d'occurrences attendues de cet événement selon une hypothèse de bruit de fond. En utilisant cette variable, un événement significatif au sens phénoménologique peut être détecté mathématiquement en tant qu'événement dit ϵ-*significatif*, c'est-à-dire dont le NFA est inférieur à ϵ (intuitivement, s'il se produit un événement qui a peu de chance de correspondre à une fausse alarme, cet événement signifie sans doute quelque chose). De nombreux problèmes de vision par ordinateur ont pu être résolus en prenant simplement ϵ égal à 1. Quand ϵ est inférieur ou égal à 1, l'événement concerné est dit *significatif*. Le principe de Helmholtz a été appliqué au problème de détection de points de fuite dans [ADV03] et dans [LGvGRM14].
Dans [ADV03], la théorie de Santaló [San04] est utilisée pour partitionner le plan image infini en une famille finie de régions dites *régions de fuite*. Des points de fuite significatifs sont alors détectés quand un grand nombre de droites (portées par des segments détectés dans l'image) se rencontrent dans une région de fuite, produisant un NFA faible. Cette méthode n'a malheureusement été évaluée que qualitativement. Toutefois, l'idée d'utiliser la théorie de Santaló pour calculer les frontières des régions de fuite est reprise dans nos travaux, bien que de manière différente.
Dans [LGvGRM14], le détecteur d'alignements de points présenté dans [LMRvG15] et lui-même basé sur le principe de Helmholtz est utilisé à deux reprises, une première fois dans le domaine image, pour regrouper des segments de droite en droites plus précises, puis une seconde fois dans le domaine dual, où des point alignés correspondent à des droites concourantes dans le domaine image. Cette méthode obtenait la meilleure précision en 2014 (94.51% pour YU, 89.20% pour EC).

2.2.3 Détection première de la ligne d'horizon

Comme nous l'avons mentionné en introduction, le principe consistant à détecter la ligne d'horizon avant de détecter les points de fuite sur cette ligne a été introduit simultanément, en 2016, dans deux contributions, celle de Zhai et al. [ZWJ16] et notre propre contribution présentée à Eurographics 2016 [15], toutes deux basées sur le même schéma : proposer des droites candidates à la ligne d'horizon, attribuer un score à chaque droite et retenir la meilleure droite selon les scores obtenus.
Dans [ZWJ16], un CNN est utilisé pour extraire un "contexte global" de l'image visant à générer un ensemble de candidats à la ligne d'horizon. Pour chaque candidat, des points de fuite potentiels sont initialisés suivant un échantillonage aléatoire couplé à une optimisation discrète, puis affinés

à l'aide d'un algorithme de type EM (plus de détails sur cette méthode sont donnés en section 2.5.3). Le score de chaque candidat à la ligne d'horizon est basé sur la consistance des segments de droite détectés dans l'image relativement aux points de fuite estimés. Cette méthode obtenait la meilleure précision en 2016 : 94.78% pour YU, 90.80% pour EC. En section 2.5, nous analysons en détails les différences en terme de résultats entre cette méthode et notre approche *a contrario*. Dans [15], la ligne zénithale est obtenue en utilisant un algorithme de force brute. Les milieux des segments perpendiculaires à la ligne zénithale sont alors projetés sur cette droite et les droites candidates à la ligne d'horizon sont sélectionnées au niveau des pics de l'histogramme des coordonnées ainsi obtenues. Des points de plus en plus espacés à mesure que l'on s'éloigne du centre de l'image (échantillonage en $\tan\theta(i)$ où $\theta(i)$ évolue linéairement entre $-\pi/2$ et $\pi/2$) sont considérés le long de chaque droite candidate puis évalués au regard de leur consistance en tant que point de fuite. Les candidats aux points de fuite sont alors sélectionnés au niveau des pics de la courbe des scores obtenus le long de la droite candidate. Le score final de chaque candidat à la ligne d'horizon est finalement la somme des deux meilleurs scores obtenus par les candidats aux points de fuite, ou le meilleur score dans le cas où un seul candidat aux points de fuite est détecté. Cette méthode est très rapide et facile à implémenter, mais ne se place qu'en milieu de peloton en terme de précision (90.40% pour YU, 85.64% pour EC). À noter toutefois que nous l'avons utilisée dans les travaux présentés aux chapitres 3 et 4, la méthode présentée ici ayant été développée postérieurement aux travaux du chapitre 3, et à peu près en même temps que ceux du chapitre 4. Dans la suite du mémoire, nous désignerons cette méthode par "méthode préalable" à la méthode présentée dans ce chapitre.

2.2.4 Contributions

Pour bâtir notre méthode, nous avons retenu le meilleur des travaux présentés ci-dessus. En particulier :

1. comme dans [ADV03, LGvGRM14], nous nous plaçons le cadre *a-contrario* issu de la théorie du Gestalt. Cependant, en décomposant la détection des points de fuite, traditionnellement réalisée dans un espace à deux dimensions – le plan image, en trois détections successives d'événements significatifs (ligne zénithale, ligne d'horizon, points de fuite) dans des espaces à une seule dimension, nous évitons la combinatoire et les temps de calculs prohibitifs qui caractérisent les méthodes *a-contrario* antérieures (par exemple, la méthode de Lezama et al. [LGvGRM14] est jusqu'à 37 fois plus lente que la notre – cf. le tableau en figure 2.14) ;
2. comme dans [ZWJ16], un grand nombre de candidats à la ligne d'horizon sont évalués. Dans [15], seules les droites correspondant aux pics de la courbe des scores étaient évaluées. À présent, nous utilisons un modèle de mélange de gaussiennes (GMM) pour générer un grand nombre de candidats supplémentaires autour des événements significatifs correspondant aux candidats à la ligne d'horizon. Cette étape est pour beaucoup (mais pas exclusivement) dans l'amélioration de la précision par rapport à [15] ;
3. comme dans [15] et [ZWJ16], les points de fuite sont sélectionnés le long des droites candidates à la ligne d'horizon. Cependant, grâce à l'exploitation du principe de Helmholtz, basée sur une modélisation fine du bruit de fond des segments contribuant aux points de fuite, nous obtenons un plus grand nombre de points de fuite significatifs (au sens mathématique du terme mais aussi au sens usuel), et surtout nettement moins de faux positifs que dans [ZWJ16] (en moyenne, 1 point de fuite erroné pour 24 corrects avec notre méthode, contre 1 pour 2 avec la méthode de Zhai et al., voir le diagramme en figure 2.14).

En outre, une spécificité de notre méthode est que la ligne zénithale elle-même est obtenue en considérant le principe de Helmholtz. Un intérêt de taille à procéder ainsi est que plusieurs candidats à cette ligne peuvent être considérés, c'est-à-dire plusieurs orientations de la ligne d'horizon. Cela permet d'éviter certaines confusions, notamment lorsque la verticale de la scène

est masquée par une autre direction, proche de la verticale (nous montrons un exemple d'une telle situation dans la description de la méthode ci-dessous).
Grâce à ces améliorations, notre méthode est à ce jour la plus précise en ce qui concerne la détection de la ligne d'horizon sur les deux jeux de données usuels (95.35% pour YU, 91.10% pour EC) sans que ne soient remises en cause ni la facilité d'implémentation, ni la rapidité d'exécution que nous soulignions à propos de notre contribution de 2016 [15]. Elle est d'ailleurs plus rapide que la méthode décrite dans [15]. Elle est aussi beaucoup plus rapide que la méthode de Lezama et al. [LGvGRM14], et légèrement plus rapide que la méthode de Zhai et al. [ZWJ16], les deux méthodes les plus précises avant 2018.
Dans la section 2.3, nous détaillons comment sont sélectionnés, selon le principe de Helmholtz, les candidats à la ligne zénithale et à la ligne d'horizon. Dans la section 2.4, nous montrons comment les hypothèses de points de fuite sont générées le long des droites candidates à la ligne d'horizon, et en particulier comment est calculé, grâce à la théorie de Stantaló, le bruit de fond des segments supposés se rencontrer en ces points. Enfin, en section 2.5, nous présentons des résultats expérimentaux et analysons ces résultats.

2.3 Candidats à la ligne d'horizon

2.3.1 Détection *a contrario* des candidats à la ligne zénithale

Comme nous l'avons déjà précisé, la ligne zénithale $\mathcal{L}_z$ est la droite reliant le point principal au zénith (point de fuite correspondant à la direction verticale de la scène). Une estimée initiale de cette droite est obtenue en considérant le fait que les segments de droite 3-D verticaux dans la scène se projettent dans l'image en des segments dont la droite portée, lorsqu'elle passe par le point principal, est confondue avec la ligne zénithale $\mathcal{L}_z$ ou, lorsqu'elle passe dans une bande étroite autour du point principal, est quasi-parallèle à $\mathcal{L}_z$ (voir la figure 2.5). Cela conduit à un gestalt de *parallélisme* du second ordre, qui peut être détecté en recherchant les *modes significatifs maximaux* (MSM – voir l'annexe A) [DMM07] d'un histogramme d'orientations. Plus précisément, la procédure utilisée pour détecter les candidats à la ligne zénithale est la suivante (figure 2.6) :

1. un ensemble de M segments d'orientations $\theta_i \in [0, \pi[$ est détecté dans l'image en utilisant l'algorithme LSD [GvGJMR12],
2. les segments éloignés du point principal [3] ou dont l'orientation est éloignée de la verticale de l'image [4] sont éliminés (la figure 2.6(A1) montre les segments retenus suite à cet élagage),
3. un histogramme d'orientations divisé en L_z boites (*bins*) de même largeur (nous dirons "un L_z-histogramme d'orientations") est calculé à partir des segments restant (figure 2.6(A2)) et les MSM de cet histogramme sont calculés (bins bleus en figure 2.6(A3)) ; les orientations correspondant au milieu du bin le plus haut de chaque MSM sont choisies comme estimées initiales des candidats à la ligne zénithale (cercles colorés en figure 2.6(A3)),
4. pour chaque estimée initiale, un ensemble de candidats aux segments verticaux (dans la scène) est sélectionné par seuillage de l'angle entre les segments de l'image et l'estimée initiale [5] (figure 2.6(B1) : les segments sont dessinés en utilisant les couleurs des cercles associés en figure 2.6(A3)) ; le point d'intersection des droites portées par ces segments (dans la direction des droites discontinues en figure 2.6(B2)) et un ensemble de segments inliers sont obtenus en utilisant un algorithme RANSAC [FB81] ; finalement, les points d'intersections (candidats au zénith) sont affinés à l'aide d'une SVD (*Single Value Decomposition*) impliquant les segments inliers uniquement.

3. $|\mathbf{l}_i^T \mathbf{c}| > d_{PP}$, où $\mathbf{l}_i$ est le vecteur des coordonnées homogènes des segments, normalisé tel que $\sqrt{l_{i1}^2 + l_{i2}^2} = 1$ et $\mathbf{c}$ est le vecteur des coordonnées homogènes du point principal.
4. $|\theta_i - \pi/2| < \theta_v$.
5. $|\theta_i - \theta_{\mathcal{L}_z}| < \theta_z$, avec $\theta_{\mathcal{L}_z} \in [0, \pi[$ l'orientation de l'estimée de $\mathcal{L}_z$ considérée.

L'étape 4 est la même que dans [ZWJ16]. Les MSM sont détectés en utilisant l'estimation de déviation large du NFA (voir l'annexe A), avec

$$p(a,b) = (b-a+1)/L \tag{2.3}$$

($L = L_z$) la probabilité *a priori* qu'un segment de droite ait son orientation dans un des bins de l'intervalle $[a,b]$ (a,b étant des numéros de bin, à valeurs entières) : une fonction de densité de probabilité (PDF) uniforme est donc utilisée comme bruit de fond de cette détection. Dans la plupart des cas un seul MSM est détecté, mais il peut arriver, comme c'est le cas en figure 2.6, que plusieurs modes soient obtenus (en moyenne, 1,71 MSM sont obtenus sur YU, 1,66 sur EC) et que le mode obtenant le NFA le plus petit ne corresponde pas à la direction recherchée. Un intérêt d'utiliser ici une approche *a contrario* est que toutes les hypothèses de ligne zénithale sont utilisées pour générer les candidats à la ligne d'horizon, de sorte que la solution correcte peut être obtenue dans les cas difficiles où l'hypothèse la plus évidente ne correspond pas à la solution attendue (figure 2.6(B2) : la ligne d'horizon attendue est représentée par la droite jaune discontinue, la ligne d'horizon estimée, très proche de la ligne attendue, par la droite cyan continue). Dans les travaux décrits dans [15] et [ZWJ16], une seule hypothèse de ligne zénithale est considérée, ce qui peut conduire à des résultats incorrects dans une telle situation (la figure 2.6(B3) montre par exemple le résultat obtenu dans cette image par la méthode de Zhai et al. [ZWJ16] : la solution est incorrecte). Plus rarement, il arrive qu'aucun MSM ne soit détecté dans l'histogramme d'orientations. Dans ce cas, la direction verticale de l'image est utilisée comme estimée initiale de la ligne zénithale, et affinée conformément à l'étape 4 de la procédure décrite ci-dessus.

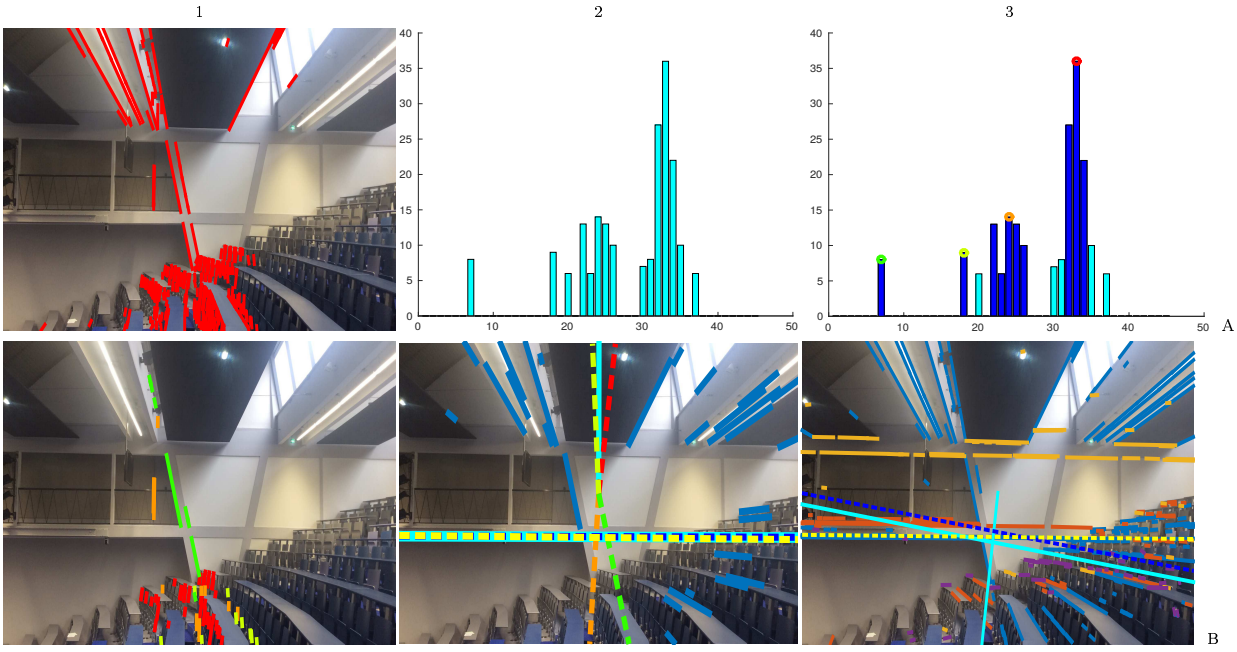


FIGURE 2.6 – Les différentes étapes de la detection *a contrario* de la ligne zénithale, décrite dans le corps du texte.

2.3.2 Détection *a contrario* des candidats à la ligne d'horizon

La détection de la ligne d'horizon est basée sur les propriétés géométriques suivantes (figure 2.5) :

1. la ligne d'horizon est perpendiculaire à la ligne zénithale,
2. n'importe quel segment de droite horizontal (dans la scène) à hauteur du centre optique de la caméra se projette sur la ligne d'horizon (est aligné avec la ligne d'horizon) quelque soit son orientation 3-D dans le plan horizontal passant par le centre optique (figure 2.4(droite)).

En utilisant ces deux propriétés, on obtient que les projections de tous les segments horizontaux de la scène situés à la même hauteur que le centre optique de la caméra sont alignés sur une droite perpendiculaire à la ligne zénithale (la ligne d'horizon). Cela induit un gestalt d'*alignement* du second ordre, qui peut être détecté en calculant les MSM d'un histogramme de coordonnées. Plus précisément, la procédure utilisée pour détecter les candidats à la ligne d'horizon est la suivante (figure 2.5) :

1. les segments non perpendiculaires à la ligne zénithale [6] sont écartés,
2. les milieux des segments restant sont projetés sur la ligne zénithale selon une projection orthogonale et leurs coordonnées, centrées par rapport à la projection orthogonale du point principal sur la ligne zénithale, sont calculées,
3. un L_h-histogramme des coordonnées obtenues est généré et les N_{init} MSM de cet histogramme sont détectés en utilisant l'équation (2.3), avec $L = L_h$, pour définir le bruit de fond (bins rouges en figure 2.5).

À nouveau, bien que plus rarement que pour la ligne zénithale, cette procédure peut conduire à plusieurs MSM (une moyenne de 1,03 MSM est obtenue sur YU, 1,06 sur EC). Les milieux des bins les plus hauts de chaque MSM sont les coordonnées le long de la ligne zénithale des N_{init} candidats à la ligne d'horizon (représentés par des lignes bleues discontinues en figure 2.5 et dans la plupart des figures concernées dans ce chapitre). Le cas où aucun MSM n'est détecté en suivant cette procédure est considéré dans la section suivante.

2.3.3 Échantillonage de candidats supplémentaires

L'estimation *a contratrio* de la ligne d'horizon peut être imprécise dans certains cas, en raison à la fois de la discrétisation de l'histogramme et de la hauteur effective des segments horizontaux 3-D utilisés pour cette estimation, qui correspond rarement précisément à la hauteur du centre optique de la caméra. Reprenant l'approche utilisée dans [ZWJ16], nous augmentons la précision de la méthode en échantillonnant des candidats supplémentaires, perpendiculairement à la ligne zénithale.

Dans [ZWJ16], la PDF utilisée pour cet échantillonage est un modèle gaussien, adapté à la distribution de probabilité des catégories obtenue en sortie du CNN. Dans notre cas, l'échantillonnage vise à affiner la position, le long de la ligne zénithale, de la ligne d'horizon obtenue comme mode de l'un des MSM détectés dans l'histogramme des coordonnées. La valeur du NFA correspondant à chaque MSM n'étant pas suffisamment fiable pour ne conserver que le mode le plus significatif, nous optons pour un GMM dont les modes sont les coordonnées de *tous* les candidats initiaux et les écarts-types égaux à une constante σH, où H est la hauteur de l'image et la valeur de σ est donnée en table 2.1. Nous tirons $S - N_{init}$ candidats supplémentaires selon cette PDF, équitablement répartis entre les N_{init} candidats initiaux.

Dans le cas où aucun MSM n'est détecté à l'issue de la procédure décrite dans la section précédente, nous n'avons aucune idée *a priori* de la position de la ligne d'horizon le long de la ligne zénithale. Cela peut se produire lorsqu'il n'y pas suffisamment de segments à hauteur de caméra, ou lorsque la caméra est inclinée de telle manière que la ligne d'horizon sort des frontières de l'image. Les coordonnées des S candidats à la ligne d'horizon sont alors échantillonnés linéairement entre $[-2H, 2H]$.

Notons que, dans les deux cas, GMM ou échantillonage linéaire, des candidats sont tirés "un peu partout", à l'intérieur comme à l'extérieur de l'image. La présence de segments horizontaux à hauteur de caméra n'est donc pas un prérequis *sine qua non* de notre méthode. Elle permet en revanche d'obtenir des résultats *plus précis* lorsqu'elle est effective. De fait, cette présence est en pratique fréquemment tangible en environnement "fabriqué par l'homme", urbain, intérieur ou même industriel. Il se trouve que des objets d'intérêt tels que des posters ou tableaux en intérieur, panneaux de signalisation en extérieur etc. sont généralement placés à hauteur de vue afin d'être remarqués par l'homme. En extérieur, les toits des véhicules, les bords des fenêtres du

6. $||\theta_i - \theta_{\mathcal{L}_z}| - \pi/2| < \theta_h$.

rez-de-chaussée etc. sont également à peu près situés à hauteur de vue, tout du moins proches de la ligne d'horizon dans l'image lorsqu'ils sont observés depuis une distance suffisamment éloignée. À nouveau, cette présence n'est pas indispensable à la détection de la ligne d'horizon. Nous présentons d'ailleurs en section 2.5 les résultats obtenus par notre méthode en utilisant uniquement l'échantillonnage linéaire : la méthode reste au premier rang sur YU, et ne descend qu'au troisième rang (est donc meilleure que sept des méthodes évaluées) sur EC. L'utilisation d'une approche *a contrario* pour la détection des points de fuite de long de la ligne d'horizon, que nous décrivons à présent, est en revanche plus déterminante quant à la qualité de cette détection, comme en attestent les expérimentations présentées en troisième partie de la section 2.5.

2.4 Candidats aux points de fuite

Les S droites candidates à la ligne d'horizon sont évaluées à l'aune des points de fuite susceptibles d'être détectés le long ces droites. Supposons qu'une droite candidate $\mathcal{L}$ définie par ses coordonnées polaires (θ, ρ) corresponde effectivement à la ligne d'horizon. Les points d'intersection entre les droites portées par les segments détectés dans l'image et la droite $\mathcal{L}$ sont alors supposés s'accumuler autour des points de fuite (figure 2.7(A,B)). Dans le même esprit que pour la détection *a contrario* de la ligne d'horizon, ces accumulations peuvent être détectées en recherchant les MSM d'un histogramme des coordonnées des intersections. Cependant, la probabilité *a priori* $p(a, b)$ d'obtenir une intersection dans l'intervalle $[a, b]$ n'est dans ce nouveau problème plus linéaire en la largeur de l'intervalle. Utiliser l'expression (2.3) pour définir le bruit de fond de la détection *a contrario* des points de fuite peut donc conduire à des détections imprécises, voire incorrectes (figure 2.7(B) : le MSM obtenu en utilisant l'équation (2.3), montré en rouge, est très large et son bin le plus haut ne correspond pas à un point de fuite de l'image). Dans cette section, nous donnons la probabilité $p(a, b)$ devant effectivement être utilisée comme bruit de fond dans ce contexte, puis détaillons comment les candidats à la ligne d'horizon sont évalués suite aux résultats obtenus par cette détection (en supposant que chaque candidat est effectivement la ligne d'horizon, c'est-à-dire, d'une certaine manière, en raisonnant par l'absurde).

2.4.1 Modèle de bruit de fond

Pour simplifier le problème, nous considérons que le domaine image est un cercle $\mathcal{C}$ de centre O et de rayon 1 (figure 2.7(A)). Les coordonnées polaires des segments détectés sont supposées uniformément distribuées sur le domaine image. La probabilité $p(a, b)$ peut alors être obtenue en utilisant une propriété établie par Luis A. Santaló à la fin des années 1970 [San04] :
Si K_1, K_2 sont deux ensembles convexes bornés du plan (qui se chevauchent ou non) et L_1, L_2 sont les longueurs de leurs frontières $\partial K_1, \partial K_2$, la probabilité qu'une corde aléatoire de K_1 intersecte K_2 est donnée par :

$$p = \frac{L_i - L_e}{L_1}, \tag{2.4}$$

où L_e est la longueur de la couverture externe C_e de K_1 et K_2 et :
- *L_i est la longueur de la couverture interne C_i de K_1 et K_2 si $K_1 \cap K_2 = \emptyset$,*
- ou *$L_i = L_1 + L_2$ si K_1 et K_2 se chevauchent.*

La couverture externe C_e de K_1 et K_2 est la frontière de l'enveloppe convexe de $K_1 \cup K_2$ (voir la figure 2.8). Elle peut être interprétée comme un élastique fermé entourant K_1 et K_2. La couverture interne C_i peut aussi être considérée comme réalisée par un élastique fermé entourant K_1 et K_2 mais "vrillé" de sorte à se superposer à lui-même en un point situé entre K_1 et K_2 .
Ce résultat peut être appliqué à notre problème de la manière suivante. Soit O' la projection orthogonale de O sur la droite candidate $\mathcal{L}$ et soit X un point sur $\mathcal{L}$ à une distance signée x de O' (figure 2.9). Prenons $K_1 = \mathcal{C}$ ($L_1 = 2\pi$) et $K_2 = [O'X]$ ($L_2 = 2|x|$). D'après le résultat de Santaló, la probabilité qu'une corde de $\mathcal{C}$ (une droite portée par un segment de l'image) rencontre la droite $\mathcal{L}$ entre O' et X dépend de si la droite $\mathcal{L}$ coupe le cercle $\mathcal{C}$ ou non.

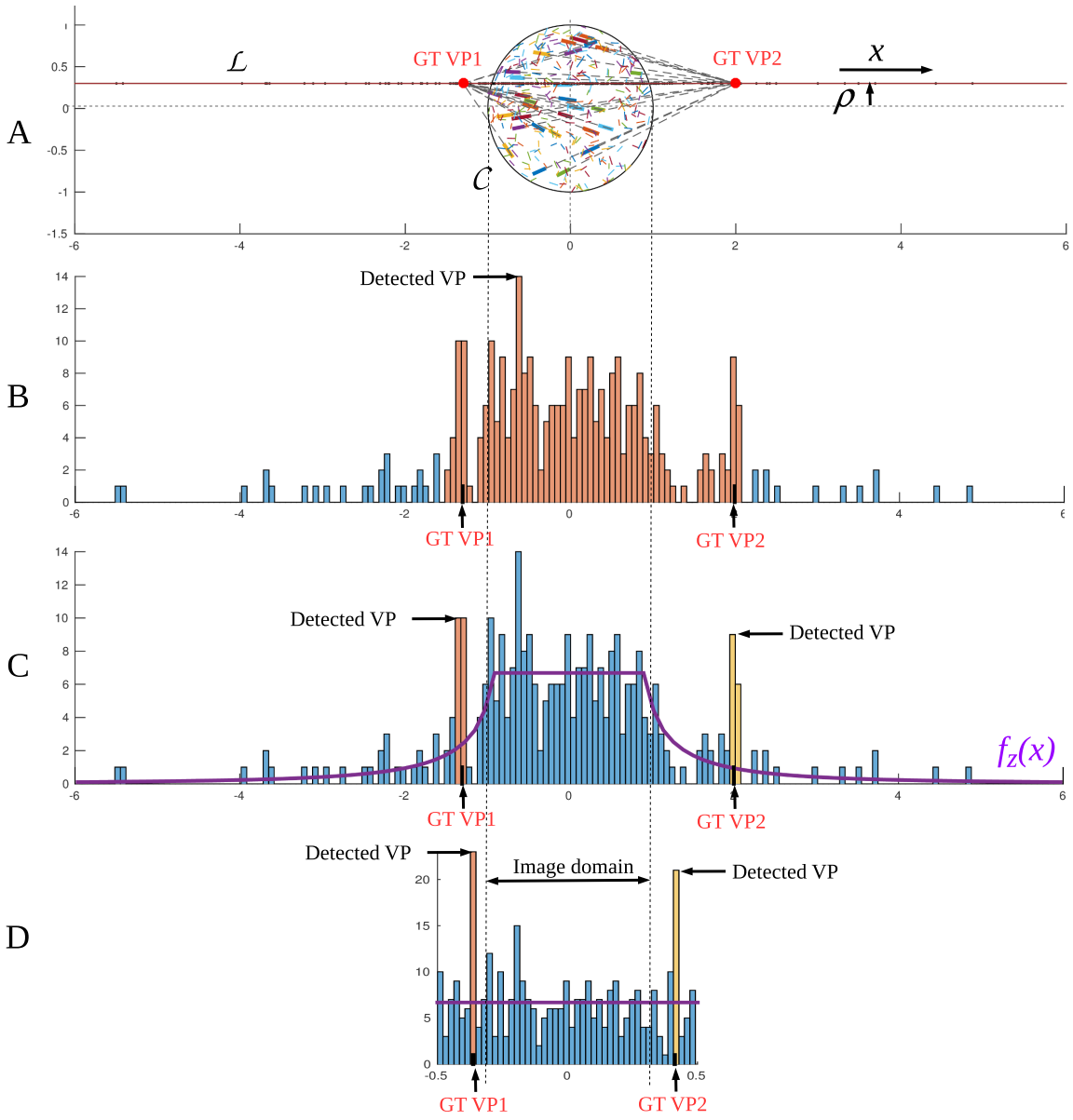


FIGURE 2.7 – Chaque segment de droite de l'image donne lieu à un point d'intersection avec la ligne d'horizon (A). Les modes significatifs maximaux d'un histogramme des coordonnées 1-D de ces intersections (en rouge et en jaune) sont sensés être détectés au niveau des points de fuite. Différents résultats sont présentés suivant le modèle de bruit de fond utilisé et la manière dont l'histogramme est généré : B : Utilisation d'un modèle de bruit de fond uniforme conduisant à une détection incorrecte des points de fuite. C : Utilisation du modèle de bruit de fond adapté au problème (courbe violette) permettant de détecter correctement les points de fuite dans un histogramme étiré. D : Application de la fonction de probabilité $p(x)$ aux coordonnées des intersections permettant d'obtenir un histogramme plus compact, dans lequel les points de fuite sont toujours correctement détectés, avec une précision identique, à l'intérieur du domaine image, à la précision obtenue précédemment.

Cas 1 : $\mathcal{C} \cap \mathcal{L} = \emptyset$ (figure 2.9(gauche)). Soient A, B (resp. C, D) les points de contact des tangentes au cercle $\mathcal{C}$ passant par le point O' (resp. X). On a :

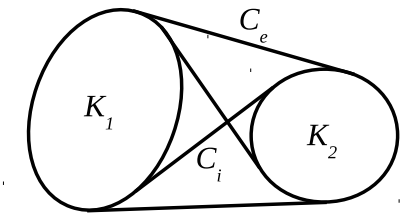


FIGURE 2.8 – Couvertures externe C_e et interne C_i de deux ensembles convexes K_1, K_2 [San04].

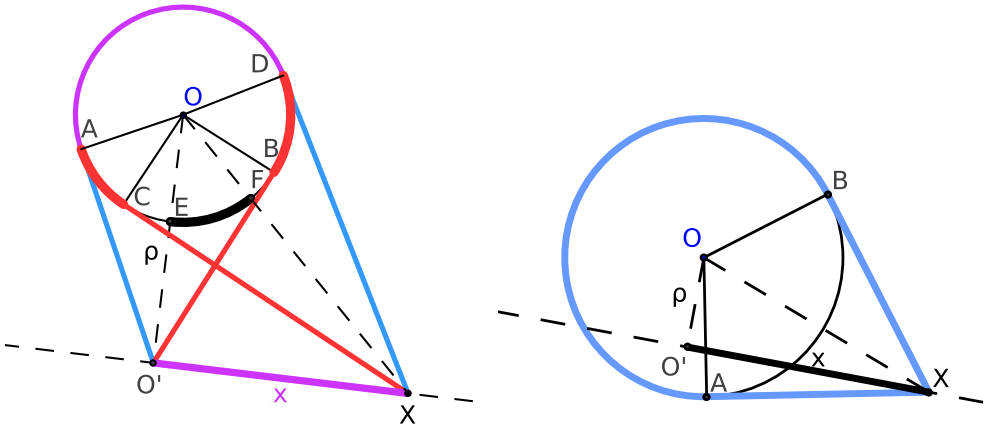


FIGURE 2.9 – Illustrations du calcul de la probabilité qu'une corde du cercle de centre O et de rayon 1 intersecte la droite $(O'X)$ entre les points $0'$ et X, suivant que la droite $(O'X)$ coupe le cercle (à droite) ou ne le coupe pas (à gauche).

$$\begin{aligned} L_e &= O'X + XD + \widehat{DA} + AO' = O'X + XC + \widehat{DA} + BO', \\ L_i &= XO' + O'B + \widehat{BD} + \widehat{DA} + \widehat{AC} + CX, \\ p &= \frac{L_i - L_e}{L_1} = \frac{\widehat{BD} + \widehat{AC}}{2\pi} = \frac{\widehat{EF}}{\pi}, \end{aligned}$$

où $\widehat{\ }$ dénote la longueur signée (positive dans le sens antihoraire) d'un arc de $\mathcal{C}$, et E, F sont les points d'intersection de $\mathcal{C}$ avec les droites (OO') et (resp.) (OX) [7]. Finalement :

$$p(x) = \frac{1}{\pi}\tan^{-1}\frac{x}{\rho}. \tag{2.5}$$

Remarque : cette expression est proche de l'inverse de la fonction d'échantillonage $s(k) = L\tan(k\Delta\theta)$ utilisée dans [15], mais dépend aussi de la distance ρ de la droite au centre de l'image, qui n'est pas prise en compte dans [15].

Cas 2 : $\mathcal{C} \cap \mathcal{L} \neq \emptyset$. Dans ce cas, nous avons $p = (L_1 + L_2 - L_e)/L_1$ où L_e dépend de si le point X est à l'intérieur ou à l'extérieur du cercle $\mathcal{C}$.

Dans le sous-cas où X est à l'intérieur du cercle, $L_e = L_1$ et

$$p(x) = \frac{x}{\pi}, \tag{2.6}$$

qui ne dépend par de ρ.

Dans le sous-cas où X est en dehors du cercle (figure 2.9(droite)), $L_e = L1 - \widehat{AB} + AX + BX$ et

$$\begin{aligned} p &= (L_2 + \widehat{AB} - 2AX)/L1 \\ &= (2|x| + 2\tan^{-1}(AX) - 2AX)/2\pi, \end{aligned}$$

7. $\widehat{BD} + \widehat{AC} = 2\widehat{EF}$ est obtenu de la manière suivante : $BD = FD - FB = CF - FB = CE + EF - FB = AE - AC + EF - FB = EB - AC + EF - FB \iff AC + BD = EB + EF - FB = EF + EF$.

où A, B dénotent les points de contact des tangentes au cercle $\mathcal{C}$ passant par X. Cela implique :

$$p(x) = \frac{1}{\pi}\left(x + \tan^{-1}\left(x\sqrt{1+\frac{\rho^2-1}{x^2}}\right) - x\sqrt{1+\frac{\rho^2-1}{x^2}}\right). \tag{2.7}$$

La figure 2.10 montre $k = 101$ tracés de $p(x)$ pour k valeurs discrètes de x variant linéairement entre 0 et 10 le long d'une droite horizontale $\mathcal{L}$. Pour chaque tracé, ρ varie de manière continue entre 0 et 3 (figure 2.7(A)). Les zones correspondant aux différents cas considérés ci-dessus sont indiquées. En dehors du domaine image, pour une valeur fixe de x (pour un tracé), $p(x)$ décroît lorsque ρ augmente et, pour une valeur fixe de ρ, la différence de $p(x)$ entre deux valeurs consécutives de x (la distance entre deux tracés) décroît. À l'intérieur du domaine image, à la fois $p(x)$ et la différence de $p(x)$ entre deux valeurs consécutives de x restent constants lorsque ρ ou (resp.) x augmentent.

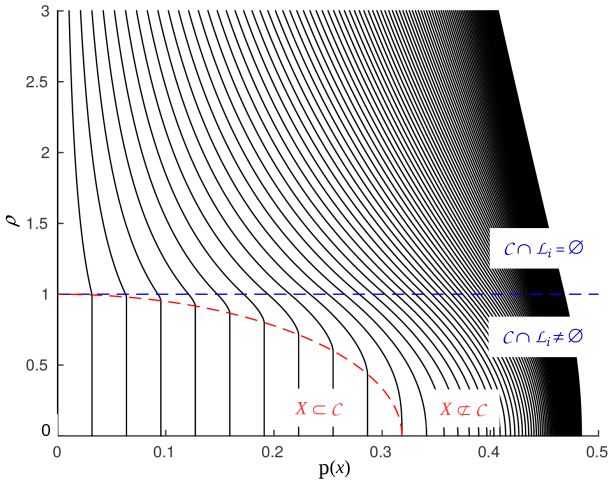


FIGURE 2.10 – Courbes montrant l'évolution de $p(x)$ pour 101 valeurs discrètes de x entre 0 et 10, en fonction de ρ variant continuement entre 0 et 3.

Finalement, étant donnés un histogramme des coordonnées des points d'intersection et un intervalle $[a, b]$ de cet histogramme, la probabilité *a priori* $p(a, b)$ qu'une intersection "tombe" dans un des bins entre a et b est donnée par :

$$p(a, b) = p(r(b)) - p(l(a)), \tag{2.8}$$

où $l(a), r(a)$ dénotent les bornes minimale et (resp.) maximales d'un bin a, et $p(x)$ est donné par l'une des équations (2.5), (2.6), ou (2.7), suivant le cas.

2.4.2 Détection *a contrario* des candidats aux points de fuite

La figure 2.7(C) montre la PDF $r(x) = \frac{\partial p}{\partial x}(x)$ (courbe violette) obtenue pour une droite $\mathcal{L}$ concernée par le cas 2. Dans cette figure, les MSM (intervalles rouge et jaune de l'histogramme) sont obtenus en utilisant le modèle de bruit de fond $p(a, b)$ défini en équation (2.8) : les deux points de fuite sont à présent correctement détectés. Toutefois, les segments de l'image proches d'être parallèles à la droite $\mathcal{L}$ intersectent cette droite en des points potentiellement très éloignés du centre de l'image. Nous pouvons donc avoir à considérer des histogrammes très étiré, présentant un nombre (variable) de bins potentiellement très élevé (pour une précision de détection convenable à l'intérieur du domaine image), ce qui peut rendre la procédure de détection des MSM coûteuse en temps de calcul. Pour cette raison, nous préférons adopter l'approche suivante :

1. fonction $p(x)$ est appliquée aux coordonnées des intersections, générant de nouvelles coordonnées, théoriquement distribuées uniformément entre $-1/2$ et $1/2$ sur l'ensemble des segments appartenant au bruit de fond,

2. un L_{vp}-histogramme (de taille fixe) est calculé à partir des coordonnées transformées, et les MSM de cet histogramme sont détectés en utilisant la prior $p(a, b)$ donné en équation (2.3), avec $L = L_{vp}$.

L'histogramme et les MSM obtenus en suivant cette procédure sont montrés en figure 2.7(D). Les deux points de fuite sont toujours correctement détectés, mais l'histogramme est beaucoup plus compact que précédemment (46 bins au lieu de 3630), pour une précision identique à l'intérieur du domaine image (30 bins). La précision est certes inférieure en dehors du domaine image, mais cela est compensé par le fait que l'erreur propagée, par exemple sur les directions 3D inférées des points de fuite, décroît à mesure que la distance entre le point principal et le point de fuite croît [8].

Finalement, les candidats aux points de fuite sont sélectionnés aux centres des bins les plus élevés de chaque MSM. Les coordonnées 1D de ces candidats sont ensuite affinées en utilisant un algorithme de type EM similaire à celui utilisé dans [ZWJ16]. Cet algorithme repose sur la mesure de consistance

$$f_c(\mathbf{v}_i, \mathbf{l}_j) = \max(\theta_{con} - |\cos^{-1}(\mathbf{v}_i^\top \mathbf{l}_j)|, 0), \tag{2.9}$$

où $\mathbf{l}_j$ est un segment dont la consistance avec un point de fuite $\mathbf{v}_i$ est mesurée. À la fin de cette procédure, nous sélectionnons les deux points de fuite les plus consistants $\{\mathbf{v}_i\}_{best}$ (ou un seul point de fuite s'il n'y a qu'un seul candidat) pour calculer le score de la droite candidate à la ligne d'horizon $\sum_{\{\mathbf{v}_i\}_{best}} \sum_{\{\mathbf{l}_j\}} f_c(\mathbf{v}_i, \mathbf{l}_j)$.

Il est important de noter que la mesure de consistance donnée en équation (2.9) est utilisée pour affiner les coordonnées des points de fuite détectés, mais pas pour les détecter. Cela représente une différence importante par rapport à la méthode de Zhai et al. [ZWJ16], qui repose entièrement sur cette mesure, à la fois pour détecter et pour affiner les points de fuite, ce qui est à l'origine d'un nombre important de fausses détections comme nous le montrons en section 2.5. Par ailleurs, la recherche mono-dimensionnelle des points de fuite présente plusieurs avantages par rapport aux approches *a contrario* précédemment proposées [ADV03, LGvGRM14], qui opéraient dans un espace bi-dimensionnel. Par rapport à [ADV03], nous évitons la procédure coûteuse de maximisation locale de la significativité des événements, ainsi que l'étape de filtrage des régions de fuite parasites, dues à des convergences artificielles de segments d'orientations différentes non associés à des points de fuite. Par rapport à [LGvGRM14], nous évitons la procédure hautement combinatoire de détection d'alignements de points dans l'espace dual 2-D, de même que le réglage délicat d'un nombre important de paramètres (tailles de rectangles, fenêtres locales, boites etc. – voir [LMRvG15] pour plus de détails).

TABLE 2.1 – Paramètres de l'algorithme. Première ligne : valeurs des paramètres (W est la largeur de l'image). Deuxième ligne : sensibilité des paramètres.

d_{PP}	θ_v	θ_z	L_z	θ_h	L_h	σ	S	L_{vp}	θ_{con}
$W/8$	22.5°	10°	45	1.5°	64	0.2	300	128	1.5°
0.0%	0.0%	−14.2%	−7.2%	−13.2%	−12.4%	0.0%	−11.4%	−6.4%	−28.7%

2.5 Résultats expérimentaux

2.5.1 Paramètres de l'algorithme

Le code source de notre méthode est distribué sous licence publique GNU Affero General [52]. Les valeurs des paramètres de l'algorithme utilisées pour nos expérimentations sont données en table 2.1. Elles ont été réglées manuellement à partir de quelques images provenant des jeux de données. Pour l'échantillonnage des candidats à la ligne d'horizon, nous avons utilisé le même

8. Comme l'angle θ entre l'axe optique et la direction du point de fuite est arc-tangentielle en la distance d entre le point de fuite et le point principal, l'erreur propagée $\partial\theta/\partial d$ est inversement proportionnelle à d^2.

nombre d'échantillons, $S = 300$, que dans [ZWJ16]. Le point principal est supposé au centre de l'image. Afin de quantifier la sensibilité de la méthode aux valeurs des paramètres, nous avons réalisé l'expérience suivante. Pour chaque paramètre p, nous exécutons l'algorithme 9 fois, en multipliant successivement la valeur de p par $\frac{1}{2}, \frac{5}{8}, \frac{6}{8}, \frac{7}{8}, 1, \frac{5}{4}, \frac{6}{4}, \frac{7}{4}, 2$, tout en laissant les valeurs des autres paramètres inchangées (les 20 premières images de YU et EC sont utilisées). Pour chaque paramètre, nous reportons la baisse relative entre l'AUC maximale et l'AUC minimale obtenues au cours des 9 exécutions de l'algorithme (dernière ligne du tableau). Les seuils de consistance θ_z, θ_h, et particulièrement θ_{con} (aussi utilisé dans [ZWJ16]) sont les paramètres les plus sensibles. Les tailles des histogrammes L_z, L_h, L_{vp} sont peu sensibles, bien que L_h soit plus sensible que les deux autres tailles. Les paramètres d_{pp}, θ_v et σ ne sont pas sensibles. Le nombre d'échantillons S n'est pas aussi sensible que ce l'on pourrait penser (de $S = 150$ à $S = 600$, la précision passe de 93.7% à 94.3%). À noter que cette expérience a été réalisée uniquement pour mesurer la sensibilité des paramètres, pas pour décider de leur valeur.

FIGURE 2.11 – Lignes d'horizon obtenues dans les images correspondant aux 1$^{\text{er}}$, 25$^{\text{ème}}$, 50$^{\text{ème}}$, 75$^{\text{ème}}$ et 100$^{\text{ème}}$ rangs percentiles de l'erreur d'horizon (colonnes 1 à 5, respectivement) sur YU (ligne A), EC (ligne B) et HLW (ligne C). La droite jaune discontinue montre la vérité terrain, les droites bleues discontinues les modes significatifs maximaux de l'histogramme calculé et la droite cyan continue la ligne d'horizon calculée. L'erreur d'horizon est affichée dans le coin en haut à gauche de chaque image de résultats. Les segments de droite contribuant aux points de fuite sont affichés en utilisant une couleur différente par point de fuite.

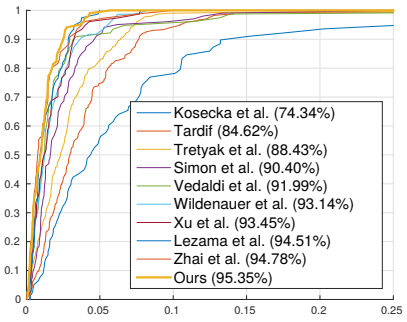
2.5.2 Précision de la ligne d'horizon

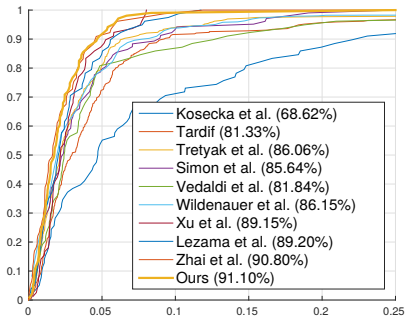
La précision de détection de la ligne d'horizon a été évaluée sur les deux jeux de données traditionnellement utilisés :

1. York Urban (YU) [DEE08], constitué de 102 images de résolution 640×480, prises en intérieur et en extérieur, et satisfaisant pour la plupart l'hypothèse de monde de Manhattan,
2. Eurasian Cities (EC) [TBKL12], constitué de 114 images de résolution 1920×1080, montrant des scènes urbaines de différents endroits du monde, sous des angles de vue variés et sans que l'hypothèse de monde de Manhattan ne soit toujours vérifiée.

Des exemples de résultats sont montrés en figure 2.11, première et deuxième lignes (resp. YU et EC). Les images sélectionnées sont celles pour lesquelles l'erreur d'horizon est la plus faible (colonne 1), la plus grande (colonne 5) ou aux 25$^{\text{ème}}$, 50$^{\text{ème}}$ et 75$^{\text{ème}}$ rangs percentiles (colonnes

2, 3, 4, resp.). La table en figure 2.12 rend compte des performances de notre méthode à l'aune de l'histogramme cumulé de l'erreur d'horizon et de l'AUC (voir la section 2.2). Nous obtenons la meilleure précision de la ligne d'horizon sur les deux jeux de données.
Sur YU, nous améliorons la précision de la méthode de Zhai et al. [ZWJ16], précédemment la plus précise, d'un gain relatif $\Delta AUC = (AUC_{new} - AUC_{old})/(1 - AUC_{old})$ de 10.9%. Ce gain de précision est important, si l'on considère notamment que le gain obtenu par [ZWJ16] par rapport à la précédente méthode la plus précise [LGvGRM14] était de 5%. Sur EC, le gain relatif par rapport à [ZWJ16] est moins important, 3.3%, mais néanmoins honorable compte tenu de la difficulté à obtenir un incrément sur ces jeux de données comportant un faible nombre d'images et considérés par de nombreux auteur depuis plusieurs années.
Afin d'analyser plus finement ces résultats, nous avons remplacé la PDF utilisée pour l'échantillonnage des candidats à la ligne d'horizon par un échantillonage linéaire entre $[-2H, 2H]$ (échantillonage par défaut lorsqu'aucun MSM n'est détecté). Les AUC alors obtenues sont indiquées dans la tableau de la figure 2.12("Samp. lin.").
La précision obtenue pour YU est similaire à celle obtenue avec la PDF basée sur les modes significatifs et supérieure à la précision obtenue avec la méthode de Zhai et al. Nous interprétons ce résultat comme lié au fait que YU est un jeu de donnée "facile" (deux grands ensembles de droites parallèles dans la scène sont détectés dans quasiment toutes les images), qui ne requiert pas d'échantillonage fin tant que ce dernier couvre l'intervalle $[-2H, 2H]$ avec une densité suffisante. Cela tend à attribuer l'amélioration de la précision sur YU par rapport à [ZWJ16] à notre procédure de notation des candidats à la ligne d'horizon (et donc de détection des points de fuite le long de la ligne d'horizon). Il apparaît en effet que la méthode de Zhai et al. obtient bien plus de points de fuite fallacieux que la notre, quelque soit le jeu de données considéré (voir la section 2.5.3).
En revanche, l'amélioration de la précision sur EC doit être interprétée différemment, puisqu'on constate que sur ce jeu de données, la PDF obtenue à partir des modes de l'histogramme, ainsi que la PDF utilisée dans la méthode de Zhai et al. permettent d'obtenir des lignes d'horizon significativement plus précises qu'avec un simple échantillonage linéaire. Il semble donc que pour ce jeu de données, la méthode d'échantillonage conjointement à la méthode de notation contribuent à l'amélioration des performances.





	YU	EC	HLW
N. images	102	103	2018
	$S = 300$		
Nous	**95,35**	**91,10**	54,43
Samp. lin.	95,24	88,42	45,59
Zhai et al.	94,78	90,80	**58,24**
	$S = 1$		
Nous	70,60	**72,78**	**45,88**
Zhai et al.	**75,89**	68,49	45,41

FIGURE 2.12 – Performances de la méthode au regard de la précision de la ligne d'horizon calculée.

La précision de la détection de la ligne d'horizon a aussi été évaluée sur *Horizon Lines in the Wild* (HLW), un jeu de données introduit récemment par Zhai et al. [ZWJ16] comprenant 2018 images de résolutions variées. Ce jeu de données n'est pas seulement plus grand que les deux autres, il est aussi plus challengeant, étant principalement constitué de photos prises par des touristes, dans lesquelles figurent des édifices, mais aussi des groupes de gens, des statues occupant une grande partie de l'image, des rochers sur fond marin, etc. De plus, les angles de roulis et de tangage des appareils photo couvrent un intervalle de valeurs très étendu, conduisant fréquemment à une ligne d'horizon éloignée des frontières de l'image, et/ou à une orientation de la ligne zénithale dépassant le seuil utilisé dans notre algorithme (voir par exemple la figure 2.11(C5)).
Des images de résultats accompagnées des valeurs d'AUC obtenues pour ces images sont montrées en figure 2.11(ligne C) et (resp.) dans la troisième colonne de la table présentée en figure 2.12.

L'approche de Zhai et al. est plus précise que la notre sur ce jeu de données, notre méthode obtenant une perte relative de 9.1% par rapport à la leur. Un résultat notable est que la précision obtenue à l'aide d'un échantillonage linéaire est ici bien plus faible que celle obtenue avec la PDF basée sur les modes significatifs (perte relative de 19.4%), ce qui semble indiquer que l'étape d'échantillonage joue un rôle crucial sur ces données. Afin d'analyser plus étroitement ce qui fait ici défaut à notre PDF d'échantillonnage comparée à celle utilisée dans [ZWJ16], nous avons testé les deux méthodes en ne considérant qu'un seul candidat à la ligne d'horizon ($S = 1$ au lieu de 300), c'est-à-dire le mode le plus significatif avec notre méthode (ou la droite horizontale dans l'image passant par le point principal si aucun mode significatif n'est détecté), et le centre de la gaussienne adaptée à la sortie du CNN pour la méthode de Zhai et al. Les résultats de ces tests sont montrés dans les deux dernières lignes de la table en figure 2.12. La précision obtenue par notre méthode est à présent quasiment la même que la précision obtenue par [ZWJ16]. Cela indique que, pour HLW, l'étalement de l'échantillonnage (les écarts-types de la mixture de gaussienne) est le facteur déterminant de la différence de performance entre notre méthode et celle de Zhai et al. Dans [ZWJ16], l'écart-type de la gaussienne d'échantillonnage est réévalué dans chaque image à partir des valeurs en sortie du CNN, tandis que nous considérons une valeur constante, empirique des écarts-types de la mixture de gaussienne (tous égaux à σ). Une manière d'améliorer nos résultats pourrait être de relier les NFA des modes significatifs maximaux aux écarts-types de la mixture.
En tout état de cause, les résultats obtenus sur HLW montrent que le pouvoir prédictif des réseaux de neurones convolutifs est intéressant pour des images difficiles à traiter par la vision par ordinateur classique, en supposant qu'une base de données étiquetées d'images similaires est disponible pour entraîner le réseau. En contrepartie, notre méthode statistique produit des résultats plus précis que la méthode utilisant l'apprentissage profond sur des images qui ne sont pas suffisamment bien représentées par la base d'entraînement. Par exemple, la figure 2.13(A1) montre une image acquise dans un environnement industriel. Notre méthode parvient à prédire correctement la ligne d'horizon et à détecter les points de fuite de l'image (figure 2.13(A1)), tandis que le CNN utilisé dans la méthode de Zhai produit une PDF d'échantillonage peu pertinente, à l'origine d'une mauvaise estimation de la ligne d'horizon et des points de fuite de l'image (figure 2.13(A2)).

2.5.3 Qualité des points de fuite

La figure 2.13(B1,B3,C1,C3) montre des exemples de résultats de détection de points de fuite obtenus par notre méthode (représentés par les segments consistants avec les points de fuite détectés). Les performances de notre méthode comparées à celles des deux précédentes meilleures méthodes [LGvGRM14, ZWJ16] ont été mesurées en comptant le nombre de bons et mauvais points de fuite obtenus sur YU et EC. Ces deux jeux de données sont représentatifs de deux résolutions d'image différentes (faible et élevée) et autorisent des détections de la ligne d'horizon plus précises qu'avec HLW, ce qui permet, autant que possible, de ne pas mélanger les deux problèmes (détection de la ligne d'horizon / détection des points de fuite).
Dans notre évaluation, un "point de fuite correct" est un point qui correspond effectivement à la convergence de droites horizontales, parallèles dans la scène, tandis qu'un mauvais point de fuite peut être qualifié de deux manière : "point de fuite erroné" s'il correspond à la convergence fortuite dans l'image de droites non parallèles dans la scène ou "point de fuite subdivisé" s'il s'agit d'un exemplaire de plusieurs points de fuite confondus, issus d'une subdivision arbitraire d'un ensemble de droites horizontales et parallèles dans la scène, sensées générer un unique point de fuite dans l'image. Dans le second cas, un "point de fuite correct" plus un "point de fuite subdivisé" par point de fuite redondant sont comptés. La figure 2.14 indique le nombre total de chacun des trois types de points obtenus sur les deux jeux de données, pour chacune des trois méthodes comparées. Notre méthode est la plus performante selon les trois critères : nous obtenons le plus grand nombre de points de fuite corrects, le plus petit nombre de points de fuite erronés et aucun point de fuite subdivisé, quelque soit le jeu de données considéré.
La méthode de Zhai et al. détecte légèrement moins de points de fuite corrects que la notre (une

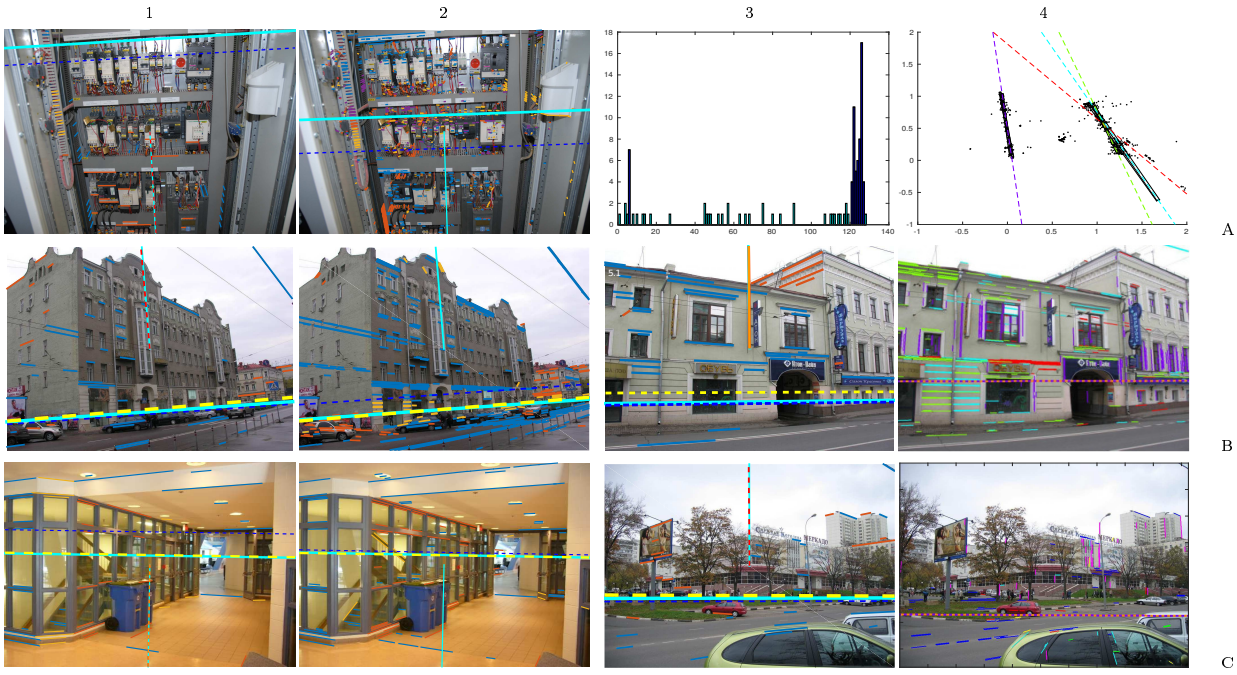


FIGURE 2.13 – Comparaisons qualitatives entre, d'une part, notre méthode (colonne 1) et la méthode de Zhai et al. [ZWJ16] (colonne 2) et, d'autre part, notre méthode (colonne 3) et la méthode de Lezama et al. [LGvGRM14] (colonne 4). Les conventions de dessin sont les mêmes qu'en figure 2.11.

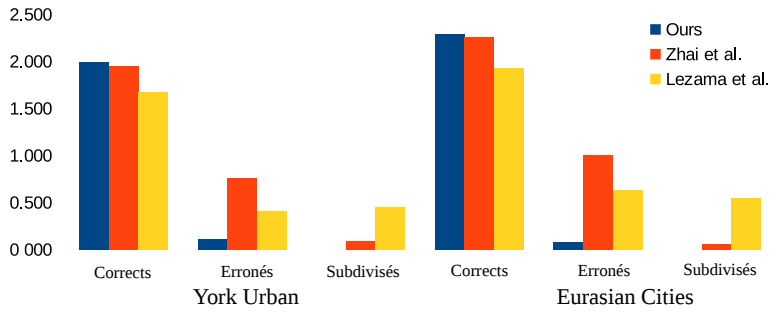


FIGURE 2.14 – Performances de la méthode au regard de la qualité des points de fuite détectés.

moyenne de 2,11 par image sur les deux jeux de données, contre 2,14 avec notre méthode) mais bien plus de points de fuite erronés, environ un pour deux points de fuite corrects, contre un pour 23 avec notre méthode. Elle obtient aussi un nombre non négligeable de points de fuite subdivisés (un pour 29 corrects, contre aucun avec notre méthode). Ces performances relativement faibles doivent essentiellement être imputées à la procédure utilisée dans [ZWJ16] pour initialiser les candidats aux points de fuite le long des droites candidates à la ligne d'horizon. Cette procédure consiste à sélectionner aléatoirement un sous-ensemble des segments de l'image $\{\mathbf{l}_j\}$ et à calculer leur intersection avec la droite candidate. Un sous-ensemble optimal de points de fuite $\mathbf{v}_i$ est extrait de ces intersections, tel que la somme des poids $\sum_{\mathbf{v}_i}\sum_{\mathbf{l}_j} f_c(\mathbf{v}_i, \mathbf{l}_j)$ est maximale, tout en s'assurant qu'il n'existe pas de points de fuite trop rapprochés dans la solution. Un seuil de distance spatiale entre deux points de fuite doit donc être utilisé, ce qui peut conduire à subdiviser des points de fuite (cf. les segments bleus et jaunes sur la façade du bâtiment en figure 2.13(B2)). De plus, la sélection aléatoire des segments peut être à l'origine de la non détection d'un point de fuite représenté par un trop faible nombre de segments (par exemple, le point de fuite consistant avec les segments jaunes en figure 2.13(C1), non détecté en figure 2.13(C2). Enfin, comme un seuil supplémentaire est utilisé pour évaluer la consistance entre points de fuite

	YU	EC	HLW
$W \times H$ (Mpixels)	0.31	0.81	1.74
Nous (S=300)	**2.07**	**2.44**	**2.88**
Zhai et al. (S=300)	2.08	2.77	3.08
Simon et al.	4.47	12.60	16.04
Lezama et al.	8.11	42.71	108.23
Nous (S=1)	**0.29**	**0.57**	**0.96**
Zhai et al. (S=1)	0.58	0.77	1.12

FIGURE 2.15 – Temps de calculs (en secondes) des méthodes évaluées.

et segments, n'importe quel ensemble de segments se rencontrant accidentellement à proximité d'un point sur la droite candidate peut générer un point de fuite erroné (cf. les segments jaunes en figure 2.13(C2)). Ces problèmes de seuillage sont naturellement écartés lorsqu'on se place dans le cadre *a contrario*, ce qui constitue l'un des principaux atouts de notre méthode.
Bien que se plaçant également dans le cadre *a contrario*, la méthode décrite dans [LGvGRM14] obtient des résultats relativement faibles sur l'ensemble des critères mesurés : le plus faible nombre de points de fuite corrects (1,80 par image), le second plus grand nombre de points de fuite erronés (un pour trois corrects) et le plus grand nombre de points de fuite subdivisés (un pour quatre corrects). Le faible nombre de points de fuite corrects (voir par exemple les points de fuite consistants avec les segments oranges en figures 2.13(B3) et (C3), non détectés en figure 2.13(B4) et (C4), resp.) peut être expliqué par le fait qu'un point de fuite peut apparaître comme significatif le long de la ligne d'horizon, mais pas dans le domaine dual de l'image complète. Le nombre élevé de points de fuite erronés (cf. les points de fuite consistants avec les segments cyan, verts, rouges et jaunes en figure 2.13(C4)) est principalement dû à la présence d'intersections accidentelles, qui se produisent plus fréquemment dans l'espace dual 2-D que le long de la ligne d'horizon. Enfin, le nombre élevé de points de fuite subdivisés est lié au fait que des points alignés dans le domaine dual (droites concourantes dans le domaine image) sont parfois relativement dispersés dans la direction perpendiculaire à l'alignement, ce qui produit plusieurs alignement significatifs ayant des orientations légèrement différentes (figure 2.13(A4,B4)). Avec notre méthode, des segments correspondant au même point de fuite peuvent se rencontrer en des coordonnées dispersées le long de la ligne d'horizon, mais généralement dans des bins contigus de l'histogramme, de sorte qu'ils se retrouvent fusionnés au sein d'un même MSM (figure 2.13(A3,B3)).

2.5.4 Temps de calculs

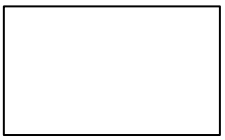
La méthode a été implémentée en Matlab et exécutée sur un ordinateur portable HP EliteBook 8570p doté d'un CPU I7-3520M. Les temps de calcul sont donnés en figure 2.15. Notre méthode est plus rapide que toutes les méthodes antérieures dont le code est disponible. De plus, contrairement par exemple à [LGvGRM14], elle n'est que légèrement affectée par un accroissement de la taille d'image, qui résulte généralement en un plus grand nombre de segments détectés. En effet, la complexité algorithmique de notre méthode étant en $O(L_z^2 + L_h^2 + S(L_{vp}^2 + M))$, celle-ci n'est que linéairement affectée par le nombre M de segments.

2.6 Conclusion et perspectives

Dans ce chapitre, nous avons décrit une méthode de détection de points fuite basée sur un schéma *a contrario*. La méthode de Zhai et al. [ZWJ16] et notre méthode ont pour point commun de générer des hypothèses de ligne d'horizon (étape 1) et de retenir l'hypothèse la plus en accord avec les points de fuite (étape 2). Ce schéma surpasse toutes les méthodes précédentes en termes de précision de la ligne d'horizon et de temps de calcul. Notre méthode montre en outre de bien meilleures performances que celle de Zhai et al. en ce qui concerne la détection des points de fuite horizontaux, ce qui peut s'avérer crucial pour certaines tâches de vision par ordinateur (par exemple, une reconstruction 3-D de la scène utilisant les directions de Manhattan). Par

ailleurs, notre méthode est utilisable dans n'importe quel type d'environnement (urbain, intérieur, industriel, ...) pour peu que des éléments architecturaux ou des objets présentant des arêtes horizontales soient présents à hauteur d'œil.
Il semble difficile d'augmenter encore significativement la précision des résultats présentés dans ce chapitre par des méthodes de vision classique [9]. En effet, la contrainte de présence d'objets fabriqués à hauteur d'œil peut être violée de deux manières : soit il n'y a effectivement pas d'objets à cette hauteur, soit l'orientation de la caméra est telle que ces objets ne sont plus visibles dans l'image (la ligne d'horizon est en dehors de l'image). Notre méthode est capable de retrouver la ligne d'horizon dans de tels cas, puisque des candidats sont tirés dans et en dehors de l'image lorsqu'aucune hypothèse n'est générée par la méthode *a contrario*. Cependant, la précision de l'échantillonnage utilisé est alors inférieure à la précision fine que l'on peut espérer obtenir avec la PDF présentée en section 2.3.3. Ce problème ne se pose pas avec les points de fuite, puisque ceux-ci peuvent se trouver en dehors de l'image tout en étant supportés par des segments de droite détectables dans l'image. Mais nous avons constaté aussi que calculer les points de fuite indépendamment de la ligne d'horizon diminue la précision moyenne.
Les résultats obtenus par la méthode de Zhai et al. sur le jeu de données HLW montrent que la ligne d'horizon prédite par régression neuronale peut être plus précise que celle prédite par la méthode *a contrario*, notamment dans les cas où la ligne d'horizon n'est pas supportée par des segments de l'image. Mais elle est aussi moins précise dans un grand nombre de cas, en particulier lorsque l'image à traiter est de nature différente de celles utilisées pour l'apprentissage (voir le cas d'un environnement industriel en figure 2.13). Les deux méthodes étant bâties selon la même architecture, il est facile de les combiner. Cependant, nos diverses tentatives en ce sens n'ont pas permis d'améliorer la précision de détection de la ligne d'horizon de manière significative sur l'ensemble des jeux de tests. Un exemple de combinaison évaluée a été de rechercher les points de fuite le long de droites candidates issues pour moitié (150 échantillons) de l'étape 1 de notre méthode (GMM centré sur les MSM) et pour moité de l'échantillonnage provenant du CNN utilisé par Zhai et al. La précision de la ligne d'horizon obtenue alors est supérieure sur HLW ($AUC = 55,74$ au lieu de $54,43$), ce qui n'est pas surprenant puisque notre méthode est peu adaptée au type d'images présentes dans ce jeu de données. En revanche, elle est légèrement inférieure sur EC ($90,74$ au lieu de $91,10$). Il semblerait en effet que dans certaines images de ce jeu de données, de faux candidats aient été introduit par la prédiction du CNN et aient "trouvé preneurs" parmi les segments de droite détectés en abondance dans ces images riches en texture.
Une autre voie d'amélioration est toutefois envisageable, qui consisterait à détecter des alignements de plus haut niveau sémantique que les points et les segments (fenêtres, voitures garées le long d'un trottoir, etc). Cette approche utiliserait l'apprentissage profond pour repérer les alignements et la vision classique pour en déduire les points de fuite. Des premiers résultats prometteurs ont été obtenus à la fin de la thèse d'Antoine Fond, en recherchant des alignements (par transformée de Hough) dans des cartes de caractéristiques d'une certaine couche (profonde) d'un CNN entraîné à détecter des points de fuite. Cette approche n'a cependant pas été approfondie pour le moment.

9. Le terme "vision classique" est communément employé en opposition aux méthode de vision basées sur l'apprentissage profond.



Se prêtant pour le rêve
De creux dans de l'épais,
D'ouvert dans de l'opaque.

Toujours fenêtre claire
Dans les prisons diverses,

Ouverture où passer
Ou du moins regarder

Et parfois vers soi-même
Plus à l'aise et plus soi

Là, de l'autre côté
Du rectangle qui s'offre.

Chapitre 3

Détection et reconnaissance des façades

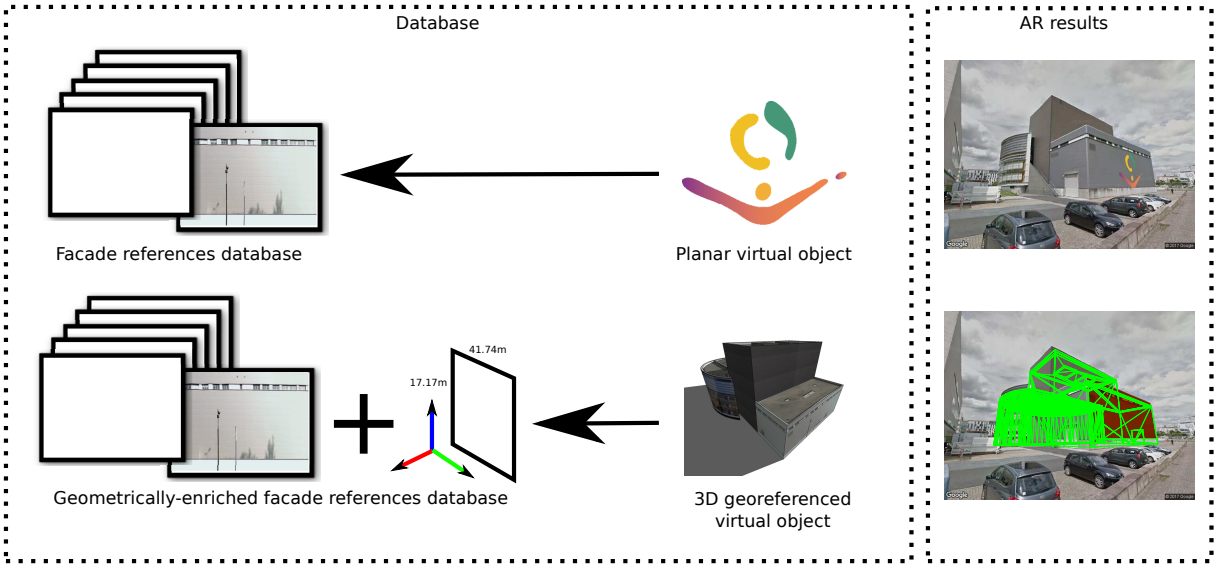


FIGURE 3.1 – Illustration de notre algorithme de détection et reconnaissance de façade appliqué à la RA. Une façade du Centre des Congrès de Nantes est automatiquement détectée et reconnue dans une vue du bâtiment (face rouge dans l'image en bas à droite). À partir de ce résultat, tout objet virtuel plan ajouté à la façade peut être déformé conformément à la transformation obtenue pour la façade (première ligne). Lorsque la façade est associée à une métrique, n'importe quel objet 3D exprimé dans le repère de la façade peut être projeté dans la vue (deuxième ligne).

3.1 Introduction

La connaissance des points de fuite de l'image est une première étape à l'estimée initiale de la pose. Connaissant les paramètres intrinsèques de la caméra (qui eux-même peuvent être obtenus à partir de points de fuite orthogonaux – voir par exemple [15], section 4.3), un point de fuite de direction horizontale et le zénith, il est aisé de rectifier l'image de manière à ce que toutes les structures planes parallèles à ces deux directions apparaissent en vue fronto-parallèle [HZ04] (notre code permettant de détecter les points de fuite [52] offre la possibilité de réaliser cette opération).
Dans l'image rectifiée, une façade peut être détectée sous forme de rectangle. Si cette façade est reconnue, sa géométrie de référence peut être transférée dans l'image requête, ainsi que n'importe quel objet infographique ajouté dans son plan (figure 3.1, ligne du haut). Si la façade est associée à une métrique respectant ses proportions, une pose peut être calculée et tout modèle 3D exprimé dans le repère de la façade (par exemple le modèle du bâtiment auquel elle appartient) peut être projeté dans l'image (figure 3.1, ligne du bas). Enfin, si le modèle 3D (ou la façade seule) est géoréférencé dans un repère plus global, il est possible de géolocaliser la caméra au sens de [APV+15, CWUF16].
Nous verrons toutefois que les coins retrouvés dans l'image coïncident rarement précisément avec les coins attendus, et la pose obtenue est généralement approximative. Suivant l'application visée, une étape de recalage fin peut donc être nécessaire pour obtenir une meilleure précision (voir le chapitre suivant). La méthode présentée dans ce chapitre a fait l'objet d'un *full paper* à ISMAR'2017 [14].

3.2 État de l'art et contributions

Détecter les façades d'une image est une tâche délicate en raison des déformations de la perspective, ainsi que de la présence fréquente de motifs répétés (typiquement, les fenêtres) et/ou d'occultations partielles des façades. Plusieurs stratégies ont été envisagées pour tenter de résoudre ce problème.

3.2.1 Détection de rectangles

Dans l'introduction du chapitre 2, nous avons brièvement décrit une première tentative visant à détecter les façades de l'image par une technique de *min-cut* rectangulaire opérant sur un ensemble de coins à angles droits. Cette technique est dans la lignée de méthodes cherchant à retrouver les bords rectangulaires des façades, en s'appuyant principalement sur les points de fuite et les contours de l'image [KZ05b, MWK08]. Dans [KZ05b], des segments de droites sont détectés automatiquement dans l'image et prolongés afin de former des hypothèses de rectangles en accord avec les points de fuite. Pour chaque hypothèse de rectangle, l'image est orthorectifiée et un histogramme de gradients (HOG) est calculé à l'intérieur du rectangle rectifié. Les hypothèses dont le HOG comporte des pics ailleurs qu'aux orientations horizontale et verticale sont rejetées. Afin d'éviter une recherche exhaustive, Micusík et al. restreignent la détection des rectangles sur une structure de voisinage donnée par une triangulation de Delaunay [MWK08]. Le problème est alors formulé en terme de maximisation de la probabilité a posteriori d'un champ aléatoire de Markov. Ces différentes méthodes permettent d'obtenir des structures rectangulaires présentes dans l'image, mais ne permettent malheureusement pas de distinguer les différents éléments architecturaux présents sur les façades (fenêtres, portes, groupes de fenêtres etc.) des façades elles-mêmes.

3.2.2 Détection d'objets

D'une portée plus large, la *détection d'objet* est traditionnellement formulée comme un problème de classification dans une fenêtre glissante. La résolution de ce problème a gagné en efficacité ces dernières années, grâce à l'apparition de techniques de proposition d'objet (*object proposal*),

dont le but est de générer rapidement un ensemble réduit de boites (ou fenêtres) susceptibles de contenir les catégories recherchées. Un classifieur est alors utilisé en seconde étape, sur l'ensemble des boites proposées. Les auteurs de [ADF12] ont été les premiers à définir le concept d'"objectivité" (*objectness*), exprimé comme un score basé sur la combinaison de multiples indices (couleur, contrastes, densité de contours etc.). Depuis lors, plusieurs technique de proposition d'objet ont été proposées. Par exemple, la méthode *Selective Search* [UvdSGS13] repose sur une segmentation de l'image en superpixels réalisée à différentes échelles, tandis que la méthode *EdgeBox* [ZD14] utilise les contours aux frontières de l'objet supposé pour calculer le score. EdgeBox semble réaliser le meilleur compromis entre rapidité et qualité, comme cela apparaît dans l'étude présentée dans [HBS14].
Comme nous l'avons déjà mentionné dans l'introduction générale, la détection d'objet a été utilisée pour aider à la reconnaissance de lieu en milieu urbain [SSJ+15]. En effet, comparer des images sur la base de l'image complète est sensible aux changements de points de vue. Comparer des régions d'images proposée par EdgeBox s'avère plus robuste à ces changements [SSJ+15]. Cependant, la méthode EdgeBox, tout comme les autres méthodes de proposition d'objet, est conçue pour détecter n'importe quel type d'objets présents dans la scène. Il résulte qu'un très grand nombre de propositions d'objets doivent être considérées, dont peu correspondent effectivement à des façades, ce qui se traduit par des temps de calcul prohibitifs et un plus grand risque de confusion dans l'étape d'appariement.

3.2.3 Classification de pixels

Enfin, toute une catégorie de méthodes utilisent l'apprentissage (profond ou non) pour classifier les pixels ou superpixels d'une image en différentes catégories. Parmi ces méthodes, [HEH05, FCNL13] obtiennent des résultats performants de classifications comportant la classe "bâtiment". Certaines méthodes sont aussi capables de détecter des sous-éléments d'un bâtiment [MMWG12, GJMG17]. Bien que prometteuses, elles ne parviennent cependant pas à distinguer deux façades adjacentes. Ces dernières années, de nouvelles architectures CNN de type *encoder-decoder* (appelées FCN ou parfois CDNN pour *fully convolutional-deconvolutional neural networks*) ont été proposées pour étiqueter des pixels (*semantic pixel-wise labeling*) [BHC15]. Ces approches obtiennent des prédictions plus lisses que celles obtenues avec [FCNL13] et sont plus rapides que [MMWG12] et [GJMG17], mais elles sont relativement imprécises, en particulier au niveau des bords des instances de classes. De surcroît, elles ne permettent pas non plus de distinguer des façades adjacentes.

3.2.4 Contributions

Nous proposons dans ce chapitre, une mesure de "façadité" (*facadeness*) appliquée à des boites de l'image, qui intègre des indices géométriques, photométriques et sémantiques et peut être évaluée rapidement. En complément des techniques de proposition d'objet existantes, nous introduisons en particulier des indices de symétrie et de répétition spécifiques aux façades. Nous proposons aussi d'utiliser des indices sémantiques basés sur la labellisation d'un réseau de type SegNet [BHC15]. Ces indices sont combinés pour générer rapidement un nombre réduit de candidats aux façades. Nous montrons que notre méthode surpasse les autres méthodes de propositions d'objets pour cette tâche particulière sur les 1000 images de la *Zurich Building Database* ainsi que sur une partie du *Cambridge Relocalisation Dataset*.
Nous démontrons l'intérêt d'une telle procédure pour la reconnaissance de façades. Cette étape est effectuée dans un système efficace qui classifie et met en correspondance les candidats en utilisant des descripteurs basés sur un réseau de neurones convolutifs. Nous prouvons que cette approche est plus robuste aux changements de points de vue et aux occultations que les méthodes de reconnaissance d'objets classiques. Enfin, nous montrons comment utiliser la détection et la reconnaissance de façades pour obtenir la pose de la caméra et augmenter la réalité.
La méthode de proposition de façades est décrite en section 3.3 et la méthode de reconnaissance de façades en section 3.4. Des résultats expérimentaux sont finalement présentés en section 3.5.

3.3 Proposition de façades

Notre algorithme de proposition de façades procède en deux étapes : un grand nombre de boites est d'abord proposé comme ensemble initial de candidats aux façades, en se basant sur les contours de l'image uniquement (section 3.3.1). Les meilleures candidats sont ensuite sélectionnés en utilisant des "indices de façadité" (section 3.3.2), dont les valeurs sont combinées à l'aide d'un perceptron multicouches (section 3.3.3).
Ces deux étapes opèrent sur des images dans lesquelles les façades ont été orthorectifiées, en utilisant la version préalable de la méthode présentée au chapitre 2. Nous supposons donc que l'hypothèse de monde de Manhattan est vérifiée et que les points de fuite correspondant aux directions du repère de Manhattan ont été détectés, ce qui permet de calculer les paramètres intrinsèques de la caméra (plus exactement, la distance focale, en supposant que le point principal est au centre de l'image) ainsi que son orientation par rapport au repère de Manhattan, comme cela est montré par exemple dans [15].

3.3.1 Candidats initiaux

3.3.1.1 Calcul simultané de la segmentation sémantique et des contours

La segmentation sémantique ne résout pas le problème de détection de façades car par exemple elle ne permet pas de distinguer deux façades adjacentes. Cependant, nous exploitons des étiquetages sémantiques de niveau pixel, d'une part pour obtenir une détection de contours plus adaptée à notre problème, et d'autre part pour calculer certains des indices de façades. Nous avons donc entraîné une version modifiée de SegNet permettant d'inférer 7 classes sémantiques (arrière-plan, façade, fenêtre, balcon, porte, ciel et route) en même temps que les contours de l'image (figure 3.2). Le fait de résoudre simultanément ces deux problèmes permet à la segmentation sémantique d'être influencée par les contours (figure 3.3), et réciproquement aux contours d'être principalement détectés aux frontières des régions sémantiques, en particulier aux bords des façades, que nous cherchons à détecter.
La base de donnée utilisée pour entraîner et tester notre réseau a été obtenue en fusionnant plusieurs bases (de manière consistante du point de vue des labels) : CMPfacadeDB [1], eTrims [2], ECP [3], INRIA [4] et labelmefacade [FRD10]. Elle contient une variété de bâtiments de styles classiques et modernes photographiés dans différentes villes européennes (Paris, Prague, Berlin, ...). La vérité terrain des contours correspond aux frontières séparant les différentes régions sémantiques. L'architecture du réseau est la même que pour SegNet, sauf que la dernière couche de déconvolution possède 9 sorties (7 pour la sémantique et 2 pour les contours) séparées en deux couches différentes. La fonction de perte utilisée pour l'entraînement est la somme pondérée de la fonction de perte logistique de ces deux couches.

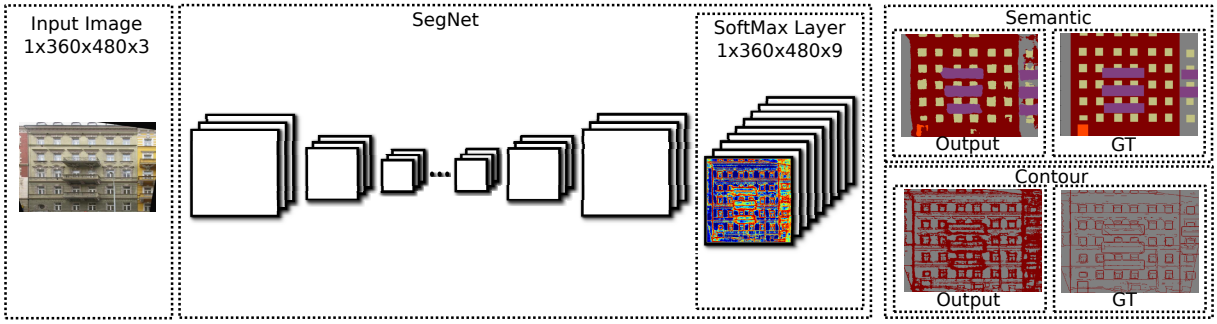


FIGURE 3.2 – Architecture de notre version modifiée de SegNet comportant deux sorties : une carte d'étiquetage sémantique et une carte de contours.

1. http://cmp.felk.cvut.cz/~tylecr1/facade/
2. http://www.ipb.uni-bonn.de/projects/etrims_db/
3. http://vision.mas.ecp.fr/Personnel/teboul/data.php
4. https://github.com/raghudeep/ParisArtDecoFacadesDataset/

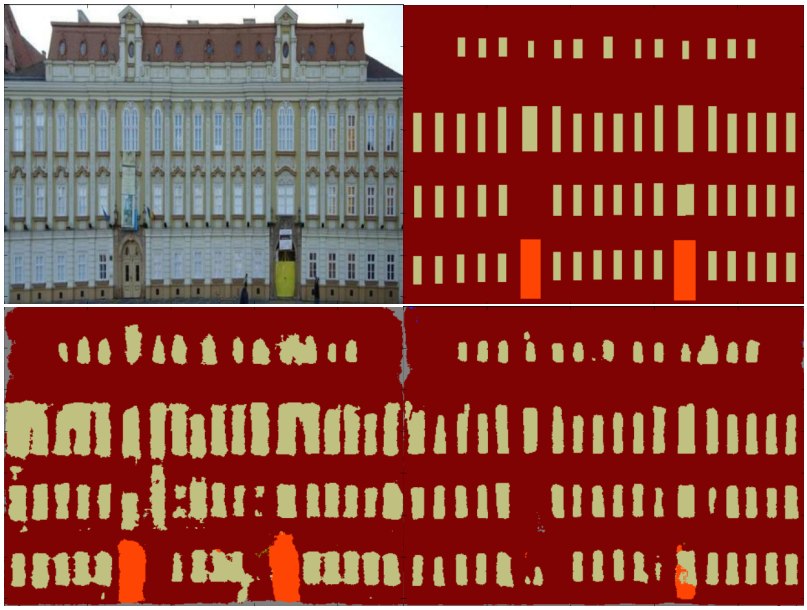


FIGURE 3.3 – Exemple de résultat de l'entraînement conjoint des cartes d'inférence sémantique et de contours. La première ligne montre une image de façade (à gauche) et la vérité terrain de son étiquetage sémantique (à droite). La seconde ligne montre la carte d'inférence obtenue par la version standard de SegNet (à gauche) et par la version modifiée (à droite). Les formes des fenêtres sont plus rectangulaire avec la version modifiée grâce à l'apprentissage conjoint de la sémantique et des contours.

3.3.1.2 Échantillonage des boites candidates

Notre principale hypothèse concernant la géométrie des façades est que celles-ci sont de forme rectangulaire. Dans les images préalablement rectifiées, nous recherchons donc des rectangles. Les bords des façades correspondant généralement à des gradients forts de l'image, nous choisissons de nous baser sur les contours détectés pour générer l'ensemble initial de candidats aux façades. Soit E la carte de points de contours obtenue en sortie du réseau SegNet modifié (figure 3.4(milieu)). Ces points sont accumulés dans deux histogrammes, l'un, H_x, correspondant aux contours projetés verticalement, l'autre, H_y, aux contours projetés horizontalement. Le produit $H_x H_y^T$ peut être vu comme une carte de vraisemblance de coins (figure 3.4(droite)). Les n maximaux locaux de cette carte sont utilisés pour générer $\frac{n(n-1)}{2}$ rectangles. Plus exactement, comme les couples de coins (haut-gauche, bas-droite) et (haut-droite, bas-gauche) définissent le même rectangle, nous retenons uniquement un ensemble de $\frac{n(n-1)}{4}$ candidats.

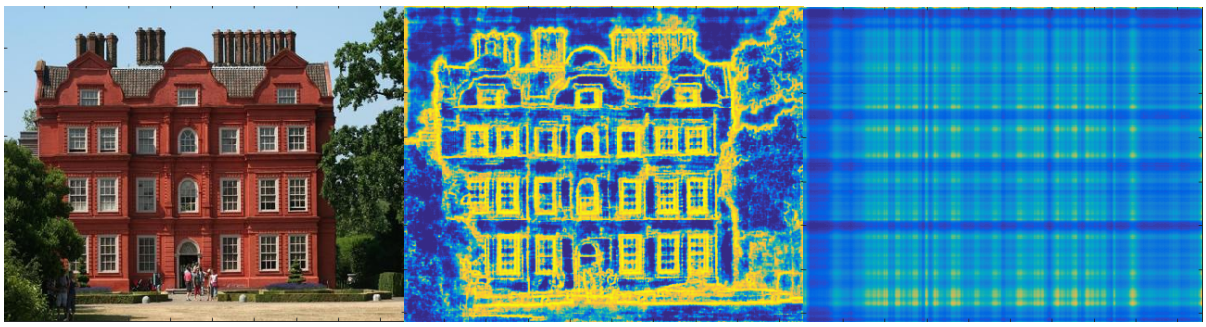


FIGURE 3.4 – Exemple d'image extraite de notre base de données (gauche). Carte de contours (milieu) et carte de vraissemblance des coins (droite) obtenues pour cette image.

3.3.2 Indices de "façadité"

Les façades de bâtiments partagent plusieurs caractéristiques visuelles. Elles sont habituellement composées de motifs rectangulaires tels que des fenêtres, des portes, des balcons etc. Ces motifs se répètent fréquemment le long de la façade, aussi bien verticalement qu'horizontalement. Les façades sont aussi très souvent symétriques, et de couleur homogène, tout du moins comparées à leur arrière-plan. Nous exploitons toutes ces caractéristiques pour décider, parmi les boites proposées lors de la première étape, quelles sont les plus susceptibles de correspondre à une façade. Pour chacune des boites candidates, nous évaluons six indices *ad hoc*. Nous réutilisons les indices de forme et de contraste de couleur définis dans [ADF12]. L'indice de contours proposé dans [ADF12] est redéfini, afin de favoriser les segments verticaux et horizontaux de l'image. Enfin, trois nouveaux indices sont introduits en vue de caractériser le contraste sémantique, la symétrie et les motifs répétés.

Ces six indices sont décrits plus en détails ci-dessous. Pour chacun d'eux, la figure 3.5, colonnes de droite, montre le meilleur rectangle (selon la valeur de l'indice) obtenu parmi tous les candidats, sur un exemple d'image de notre base. Les colonnes de gauche montrent les probabilités des valeurs d'indices d'être obtenues pour une boite correspondant (en vert) ou ne correspondant pas (en rouge) à une façade.

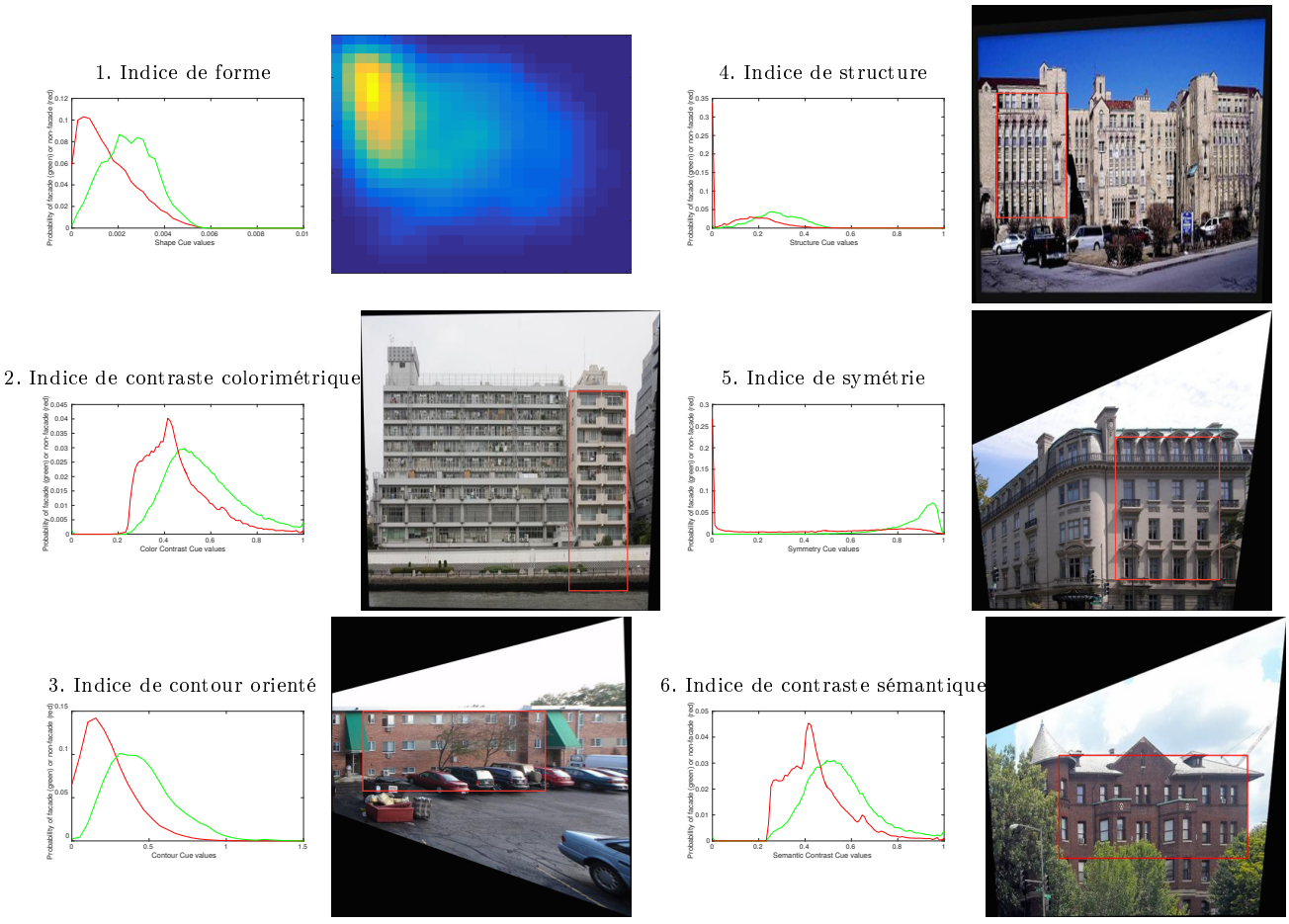


FIGURE 3.5 – Colonnes 1 et 3 : probabilités de chaque valeur d'indice d'être obtenue à l'intérieur d'un rectangle correspondant à une façade (en vert) ou à l'intérieur d'un rectangle ne correspondant pas à une façade (en rouge). Colonnes 2 et 4 : meilleur rectangle obtenu (obtenant le score d'indice considéré le plus élevé) parmi tous les candidats dans une image d'exemple de notre base d'apprentissage (pour l'indice de forme, la carte de chaleur complète de l'histogramme H est montrée).

Indice de forme Les façades sont supposées rectangulaires, mais, selon leur rapport d'aspect (hauteur/largeur), tous les rectangles n'ont pas la même probabilité d'être observés. En effet, les traditions architecturales favorisent, statistiquement, certains rapports d'aspect. Des façades extrêmement fines, par exemple, sont quasiment impossibles à rencontrer. Nous avons ainsi appris la distribution de probabilité de la hauteur et de la largeur de rectangles, correspondant à des façades, détourés à la main dans une base de 1500 images (décrite en section 3.3.3). Par soucis d'efficacité, nous utilisons une version discrétisée H, de taille 24×24, de cette distribution (figure 3.5(ligne 1, colonne 2)). Finalement, notre indice de forme correspondant au rectangle r de taille $h \times w$ est simplement :

$$s_{shape}(r) = H(h, w). \tag{3.1}$$

Indice de contraste colorimétrique La couleur en soi ne contient pas beaucoup d'information susceptible de caractériser les façades, celles-ci pouvant avoir des couleurs très différentes. En revanche, l'homogénéité de la couleur d'une façade comparée à son contexte local est un indice pertinent, comme cela a été montré dans [ADF12]. La différence des distributions de couleur entre l'intérieur du rectangle et une bande entourant le rectangle peut en effet contribuer à détecter des façades (voir par exemple la figure 3.5(ligne 2, colonne 2)). Nous prenons donc comme indice de contraste colorimétrique :

$$s_{color}(r) = 1 - \exp\left(-d_{\chi^2}\left(H_c^{b(r,\beta)}, H_c^r\right)/\sigma_c\right), \tag{3.2}$$

où H_c^r et $H_c^{b(r,\beta)}$ sont respectivement l'histogramme de couleur de l'intérieur du rectangle r et l'histogramme de couleur d'une bande d'épaisseur β entourant r. Nous utilisons l'espace de couleurs LAB, quantifié en $256 = 4 \times 8 \times 8$ bins.

Indice de contour orienté Comme les façades ont des formes rectangulaires dans les images rectifiées, nous nous attendons à avoir des valeurs élevées de gradients d'image le long des bords d'une boite correspondant à une façade. Tenant compte de cette observation, nous prenons comme indice de contour orienté :

$$s_{cont}(r) = \frac{1}{2\alpha(l+h)}\left(\sum_{b_h(r,\alpha)\cup b_b(r,\alpha)} E_x + \sum_{b_g(r,\alpha)\cup b_d(r,\alpha)} E_y\right), \tag{3.3}$$

où α est la largeur de la bande $b_x(r,\alpha)$ centrée en haut ($x = h$), en bas ($x = b$), à gauche ($x = g$) ou à droite ($x = d$) du rectangle r, et E_x, E_y sont les images binaires des contours horizontaux et (resp.) verticaux obtenus en sortie du réseau SegNet modifié.

Indice de structure Les fenêtres et les balcons se répètent fréquemment le long des directions verticales et horizontales des façades. Soit l'histogramme normalisé de 32 bins H_x^r (resp. H_y^r) des coordonnées horizontales (resp. verticales) des pixels étiquetés "fenêtre" ou "balcon". L'autocorrélation de chacun des signaux H_x^r et H_y^r est éparse (concentrée au niveau des pics) si ces signaux contiennent des répétitions marquées. Pour cette raison, nous définissons l'indice de structure de la manière suivante :

$$s_{struc}(r) = W(r)\left(\frac{\sum_{pics} R(H_x^r)}{\sum R(H_x^r)} + \frac{\sum_{pics} R(H_y^r)}{\sum R(H_y^r)}\right), \tag{3.4}$$

où $R(f) = \mathcal{F}^{-1}|\mathcal{F}(f)|^2$ est l'autocorrélation de f, $\mathcal{F}$ et $\mathcal{F}^{-1}$ étant respectivement la transformée de Fourier et la transformée de Fourier inverse d'un signal mono-dimensionnel, *pics* sont les positions des maximaux locaux du signal et $W(r)$ est le nombre de labels "façade", "fenêtre", "balcon", "porte" obtenus à l'intérieur du rectangle r, normalisé par l'aire de r.

Indice de symétrie Les façades ont souvent un axe de symétrie imparfait. Nous cherchons à définir une métrique évoluant continûment avec l'aspect symétrique de la façade. Par exemple, la corrélation croisée entre les intensités lumineuses de la moité gauche et de la moitié droite de la boite peut être très faible pour une petite asymétrie. Pour cette raison, nous proposons de subdiviser la boite en 16 parcelles (*patchs*). Pour chacun des patchs, nous calculons un descripteur HOG à partir des contours obtenus en sortie du réseau. Nous évaluons ensuite les 8 distances entre les patchs de gauche et les patchs de droite positionnées symétriquement par rapport aux patchs de gauche :

$$s_{sym}(r) = \exp\left(-\sum_{i=1}^{4}\sum_{j=1}^{2}\frac{d_{\chi^2}\left(H_e^{sym}(s(i,j)), H_e(i,j)\right)}{8\sigma_s}\right), \tag{3.5}$$

où $H_e(i,j)$ est le descripteur HOG à 8 bins du patch (i,j), $H_e^{sym}(i,j)$ est la version en miroir du vecteur $H_e(i,j)$, s est la symétrie axiale d'axe vertical (passant par le centre de l'image) et d_{χ^2} est la distance du χ^2.

Indice de contraste sémantique Les façades sont composées de primitives sémantiques telles que des fenêtres, balcons, portes, murs. La proportion de ces éléments sur une façade diffère généralement de leur proportion dans une région entourant la façade, celle-ci pouvant du reste inclure d'autres primitives sémantiques telles que des portions de ciel ou de route. Nous utilisons donc comme indice de façade supplémentaire un indice de contraste sémantique défini comme suit :

$$s_{sem}(r) = 1 - \exp\left(-d_{\chi^2}\left(H_s^{b(r,\gamma)}, H_s^r\right)/\sigma_{sc}\right), \tag{3.6}$$

où H_s^r et $H_s^{b(r,\gamma)}$ sont respectivement l'histogramme des labels sémantiques à l'intérieur de la région r et à l'intérieur d'une bande de largeur γ autour de r.
Le calcul de tous ces indices est en temps constant pour un rectangle grâce à l'utilisation d'intégrales d'image. Cette astuce est détaillée dans [VJ01] pour le calcul de sommes dans des régions et le calcul d'histogrammes locaux. La quantification des histogrammes et le nombre de patchs ont été choisis de manière à obtenir un temps constant raisonnable (inférieur à 10^3 opérations). Les largeurs de bandes α, β, γ ont été apprises à l'aide de notre base d'entraînement (voir la section 3.3.3) de manière à maximiser la séparabilité entre les distributions de probabilité positives et négatives des valeurs d'indices (figure 3.5(colonnes 1 et 3)). Les valeurs optimales de ces paramètres sont respectivement 5%, 30% and 20% des dimensions des rectangles. σ_c, σ_s, σ_{sc} sont des écart-types des distances obtenues sur les indices de couleur, de symétrie et (resp.) de contraste sémantique.

3.3.3 Combinaison des indices

Le fait que les distributions de probabilité des valeurs d'indices obtenues pour les façades et (resp.) les non-façades se recouvrent partiellement (figure 3.5(colonnes 1 et 3)) signifie qu'aucun de ces indices ne suffit à lui seul à distinguer les rectangles correspondant à une façade des rectangles ne correspondant pas à une façade. Nous considérons donc l'ensemble de ces indices à l'aide d'un perceptron multicouche. Ce perceptron est composé de deux couches cachées de 8 neurones. Il a été appris à partir d'une base de 1500 images orthorectifiées provenant de Google Street View et ImageNet, dans lesquelles les boites englobantes des façades (vérité terrain) ont été définies manuellement. Les rectangles positifs et négatifs utilisés pour cet apprentissage ont été extraits de l'ensemble des rectangles générés par la procédure d'échantillonage décrite en section 3.3.1, appliquée à toutes les images de la base d'apprentissage. Pour décider si un rectangle est positif ou négatif, nous avons utilisé la métrique classique "Intersection sur Union" (score IoU s_{IoU} [ZD14]). Un seuil de 0.5 sur l'IoU est généralement employé dans la littérature pour décider si deux régions d'image coïncident ou non. Une illustration dans [ZD14] montre par ailleurs qu'un score de 0.5 correspond à un recouvrement déjà relativement élevé. Pour ces raisons, nous

considérons comme exemples positifs les rectangles qui se superposent à un rectangle de la vérité terrain avec $s_{IoU} \geq 0.5$, tandis que les exemples négatifs sont ceux pour lesquels $s_{IoU} < 0.5$.
La valeur obtenue en sortie du perceptron peut être vue comme un score de probabilité que le rectangle considéré corresponde à une façade. Les candidats sont triés selon ce score et, tout comme dans [ADF12], un algorithme glouton est utilisé pour conserver les rectangles ne recouvrant pas significativement ($s_{IoU} \geq 0.5$) un rectangle mieux classé.

3.4 Reconnaissance de façades

Le problème que l'on cherche à résoudre à présent est le suivant : étant donnée une image acquise dans un lieu connu, nous voulons *reconnaître* les façades les plus saillantes de cette image. Par reconnaître, nous entendons à la fois *détecter* les façades sous forme de boites englobantes et *identifier* ces façades parmi un ensemble de façades, représentées sous forme d'images de référence (rectifiées) associées à ce lieu.
Ce problème, différent (complémentaire) de celui de la reconnaissance de lieu, ouvre la voie à des applications de réalité augmentée telles que l'affichage d'informations pratiques ou culturelles par-dessus les bâtiments ou le guidage de l'utilisateur vers un bâtiment qu'il cherche à atteindre. Dans ce contexte, une localisation obtenue par un GPS peut ne pas être suffisamment précise pour obtenir un détourage adéquat, mais permettre de réduire considérablement le nombre de façades à examiner lors de la phase d'identification.
Notre méthode de reconnaissance de façade comporte les trois étapes suivantes (figure 3.6) :

1. **Générer un faible de nombre de boites candidates aux façades** à l'aide de notre méthode de proposition de façades.
2. **Classifier ces candidats en "façade" ou "non-façade"** à l'aide d'un réseau de neurones prenant des descripteurs SPP (*Spatial Pyramid Pooling* [HZRS14]) en entrée. L'utilisation de ces descripteurs permet d'appliquer les filtres de convolution une seule fois pour l'image entière, au lieu de les appliquer à chaque boite proposée.
3. **Apparier les candidats classifiés "façade" aux façades de référence** associées au lieu courant, en utilisant une métrique sémantique apprise à l'aide d'un réseau siamois prenant les descripteurs SPP en entrée.

Nous détaillons à présent les étapes 2 et 3 de cette méthode.

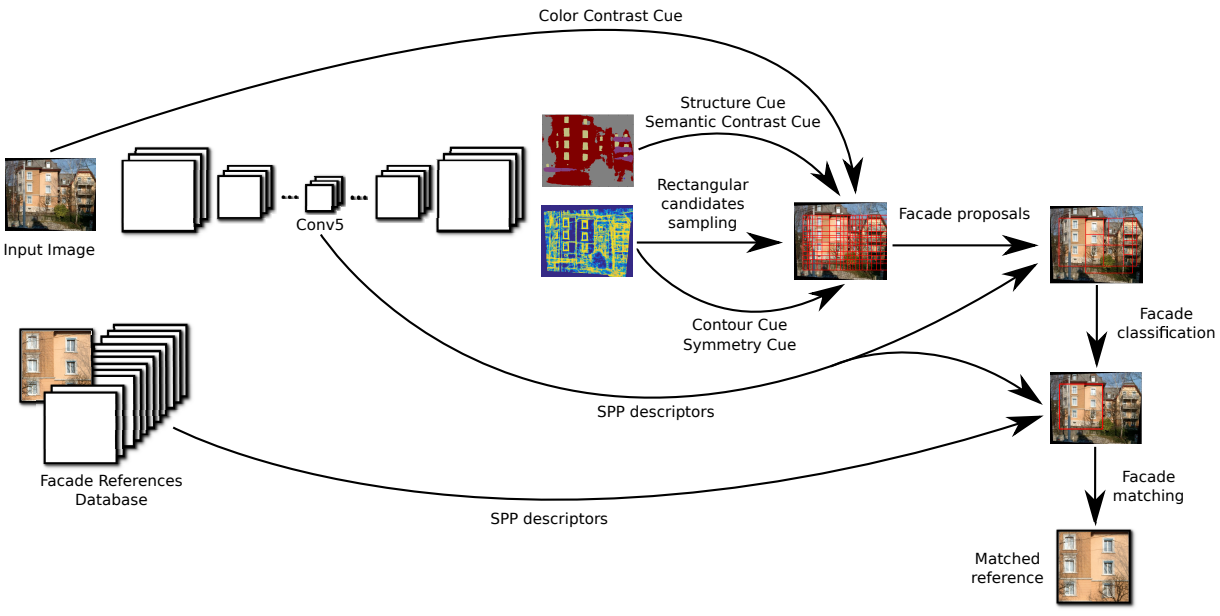


FIGURE 3.6 – Vue d'ensemble de notre méthode de reconnaissance de façades.

3.4.1 Classification en façade / non-façade

La classification de façade présente une difficulté supplémentaire par rapport à la classification d'objets classique, qui est qu'une sous-façade (par exemple, le premier étage d'un bâtiment) possède généralement toutes les caractéristiques visuelles d'une façade. Pour éviter de classifier les sous-façades en façades, il est nécessaire de considérer le contexte visuel environnant la boite candidate. Une façade (complète) doit ressembler à une façade à l'intérieur de la boite, mais pas autour de la boite. Nous proposons donc d'utiliser comme descripteur la concaténation d'un descripteur SPP calculé à l'intérieur de la boite avec un descripteur SPP calculé sur une bande entourant la boite.
Le descripteur SPP correspondant à l'intérieur de la boite est calculé à partir de la 5ème couche de convolution (`conv5`) du réseau SegNet modifié. Nous utilisons une pyramide spatiale à 3 niveaux (1×1, 2×2, 4×4) pour mutualiser les valeurs des 512 cartes de caractéristiques (*feature maps*) de la couche `conv5` (voir la figure 3.7). La résolution de `conv5` est cependant trop faible (23×30) et la bande englobante trop fine (25 % des dimensions de la boite) pour pouvoir appliquer ce schéma multi-résolution à la bande. Notre solution consiste à diviser la bande en quatre parties (haut, bas, gauche, droite) et, pour chaque partie, à utiliser une pyramide spatiale à deux niveaux (1×1, 1×4). Le descripteur SPP correspondant à l'intérieur de la boite a ainsi pour dimension $512 \times (1 + 4 + 16) = 10752$, tandis que le descripteur SPP correspondant à la bande a pour dimension $512 \times 4 \times (1 + 4) = 10240$.

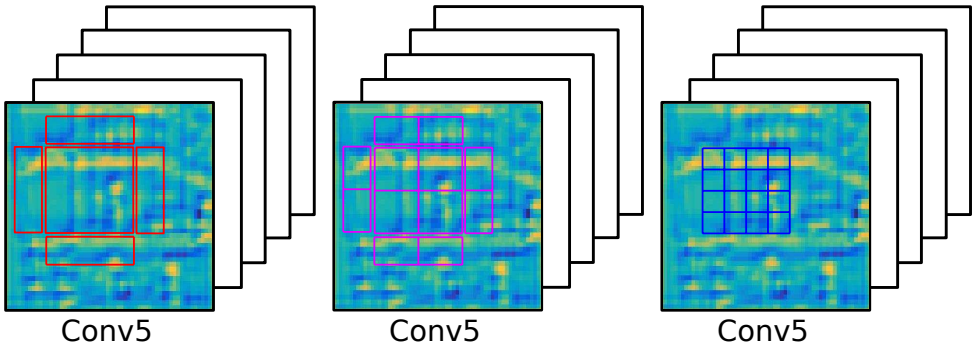


FIGURE 3.7 – Schéma de *Spatial Pyramid Pooling* sur trois niveaux (le premier en rouge, le second en magenta et le dernier en bleu) du descripteur SPP sur la couche `conv5`.

Le vecteur concaténé de dimension 20992 constitue l'entrée d'un classifieur par réseau de neurones. Ce classifieur est composé de deux couches cachées complètement connectées de taille 4096. Le réseau est entraîné sur notre base d'entraînement, à laquelle nous avons ajouté des images synthétiques. Ces images correspondent à des images de façades complètes venant d'ImageNet et de Google Street View, collées sur des images d'environnement urbains.

3.4.2 Appariement de façades

Pour l'étape d'appariement, nous pourrions utiliser le descripteur SPP correspondant à l'intérieur de la boite, comparable (à l'aide d'une simple distance euclidienne par exemple) aux descripteurs SPP calculés sur les images de référence. Cependant, distinguer deux façades différentes tout en étant robuste aux changements d'apparence d'une même façade peut être difficile. Ce problème est en fait similaire à celui de la classification à grain fin (*fine-grained classification*) qui se résout classiquement par l'apprentissage d'une métrique de similarité spécifique. C'est cette approche que nous suivons ici en utilisant un réseau de neurones siamois [CHL05]. L'idée est de trouver une fonction qui envoie les images dans un espace de dimension faible où la distance euclidienne entre points de la même classe est faible et celle entre points de classes différentes est importante. Ici, des paires de descripteurs SPP internes (a_n, b_n) sont utilisées comme entrées d'un réseau de neurones Φ à deux couches cachées de taille 2048 entraîné en mode siamois [CHL05] sur un tiers de la base ZuBuD avec la fonction de coût suivante :

$$L = \frac{1}{N}\sum_{n=1}^{N} yd^2 + (1-y)\max(m-d,0)^2, \tag{3.7}$$

avec $d = ||\Phi(a_n) - \Phi(b_n)||_2$, m la marge négative, et $y = 1$ si la paire est positive $y = 0$ sinon.
Les paires positives sont générées à partir des candidats positifs ($s_{IoU} \geq 0.5$) des différentes vues d'une même façade tandis que les paires négatives sont générées à partir de candidats positifs de façades différentes. L'espace induit par le réseau de neurones siamois est ainsi de plus petite taille et ajusté pour distinguer des façades qui peuvent sembler proches visuellement.
Nous calculons enfin la distance euclidienne entre le vecteur obtenu en sortie de ce réseau avec la boite détectée et les vecteurs obtenus en sortie (pré-calculés) avec les façades de référence. Pour chaque détection, nous choisissons le plus proche voisin dans la base de référence. Pour nous assurer que cette mise en correspondance est correcte nous imposons que l'association par plus proche voisin soit réciproque.

3.5 Résultats expérimentaux

3.5.1 Proposition de façades

Notre méthode de proposition de façades est évaluée sur la *Zurich Buildings Database* (ZuBuD) [5]. Ce jeu de données est composé de 1000 photographies de rues de Zurich prises par des piétons. Elle est divisée en 200 scènes, chaque scène se concentrant sur un bâtiment. Les changements de point de vue sont sévères et il n'est pas rare que des façades soient occultées par des arbres, des lampadaires ou des lignes électriques. Cette base comporte aussi une bonne diversité de bâtiments possédant des architectures variées, allant du style classique européen à des styles plus modernes.
Nous utilisons les images orthorectifiées de ZuBuD, dans lesquelles les façades ont été détourées manuellement. Nous comparons notre approches aux méthodes *Selective Search* [UvdSGS13], *Objectness* [ADF12] et *EdgeBox* [ZD14] en utilisant trois critères : rappel, précision et temps (Table 3.1).

TABLE 3.1 – Résultats concernant la proposition de façades.

	Selective Search	Objectness	EdgeBox	Nous
Rappel ($n = 100$)	74.89	74.18	70.81	85.16
Précision (s_{IoU})	0.59	0.63	0.61	0.68
Temps (secondes)	0.28	1.42	0.35	0.42

Le rappel est l'une des mesures de performance les plus importantes pour une méthode de proposition d'objets. Il correspond au taux d'objets retrouvés parmi les n premières propositions. La figure 3.8 montre le taux de rappel obtenu sur ZuBuD en fonction du nombre de candidats proposés. Notre méthode est évaluée sous trois configurations différentes : en considérant uniquement les indices spécifiques à notre problème (contour, structure, symétrie et contraste sémantique), en considérant uniquement les indices déjà définis dans [ADF12] (contour, forme, contraste colorimétrique) ou enfin, en considérant l'ensemble de ces indices (ces trois configurations sont désignées respectivement par *Our (problem-specific cues)*, *Our (Objectness inspired cues)* et *Our (all cues)* en figure 3.8). Un rectangle de la vérité terrain est considéré comme trouvé lorsqu'au moins un des candidats sélectionnés le recouvre avec un score $s_{IoU} \geq 0.5$.
Si la plupart des méthodes comparées obtiennent un taux de rappel de plus de 80 % avec $n = 500$, notre méthode s'avère plus performante que les autres pour n plus petit. En pratique, nous utilisons les 100 premières propositions obtenues, avec un taux de rappel de 85 %, bien

5. `http://www.vision.ee.ethz.ch/showroom/zubud/`

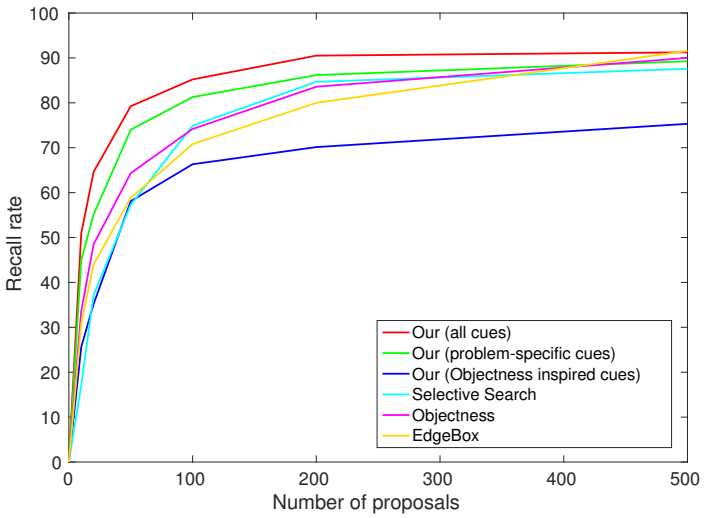


FIGURE 3.8 – Taux de rappel obtenus pour plusieurs méthodes en fonction du nombre de candidats.

supérieur aux taux de rappels obtenus avec les autres méthodes (voir la table 3.1). Nous pouvons ainsi considérer beaucoup moins de propositions qu'en utilisant les méthodes précédentes, pour la même performance et à peu près les mêmes temps de calcul. Cela peut aisément s'expliquer par le fait que nous utilisons des indices plus discriminants sur des façades que sur des objets quelconques.

La précision est calculée ici comme la valeur moyenne s_{IoU} de toutes les propositions qui se superposent à la vérité terrain avec $s_{IoU} \geq 0.5$ (nous prenons ici $n = 100$). Cette mesure n'est pas critique pour la proposition d'objets, mais dans le cas d'une détection de façades utile au calcul de pose, il est important de localiser le plus précisément possible les bords des façades. Nos meilleurs résultats de précision comparés aux autres méthodes peuvent être expliqués par le fait que nous utilisons des indices basés sur les contours des façades et le contexte environnant.

Le temps de calculs est une autre mesure de performance importante pour la proposition d'objets. Tous nos indices sont calculés en temps constant, ce qui nous permet d'obtenir des temps de calcul inférieurs ou du même ordre que ceux des autres méthodes. Ce temps est compatible avec ce que l'on attend d'une méthode d'initialisation de pose pour des applications de réalité augmentée (il est cependant trop important pour envisager d'utiliser cette méthode pour du suivi de caméra temps réel). Le code utilisé a été écrit en Matlab avec des parties critiques écrites en C. Les temps de calcul montrés en table 3.1 sont les temps moyens obtenus pour $n = 100$ sur un processeur I7-3520M associé à une carte graphique Nvidia TITAN X.

3.5.2 Reconnaissance de façades

Le jeu de données utilisé pour évaluer notre méthode de reconnaissance de façade est constitué des deux tiers d'images de la base ZuBuD non utilisées pour l'apprentissage de la métrique (section 3.4.2), dans lesquelles nous avons étiqueté manuellement 937 façades. Ces façades sont regroupées en 171 classes, chaque classe représentant une façade unique, observée sous différents angles. Les images des façades sont orthorectifiées, avec pour conséquence des artefacts de rectification et des résolutions d'image variées (figure 3.9). De plus, en raison de la présence d'occultations, certaines façades peuvent correspondre à une partie seulement d'une façade représentée entièrement dans une autre image, et des occultations peuvent subsister dans certaines images. En plus de cette diversité intra-classe, différentes classes peuvent avoir une apparence similaire. L'image de

référence de chaque classe est choisie visuellement pour sa proximité avec la vue frontale et une faible présence d'objets occultants. Les autres images de la classe sont utilisées pour les tests.

FIGURE 3.9 – Illustration de la diversité d'apparences d'une façade au sein de sa classe, en raison d'artefacts de rectification, de la présence d'occultations et d'observations partielles.

La procédure de reconnaissance de façade est évaluée en soi, et comparée à d'autres méthodes d'appariement de la manière suivante. Pour chaque approche évaluée, nous exécutons notre procédure de proposition de façade avec $n = 100$ candidats. Pour chaque image de test, nous sélectionnons le candidat qui recouvre au mieux la façade selon le score s_{IoU}. De cette manière, les candidats aux façades sont les meilleurs que nous puissions obtenir, indépendamment des performances de l'étape de détection évaluée précédemment. Pour chacun des candidats retenus, nous calculons leur descripteur SPP et recherchons leur plus proche voisin, au sens de la distance euclidienne, parmi les descripteurs des 171 façades de référence. Un appariement est considéré comme correct si la classe du plus proche voisin correspond à la classe vérité terrain de l'image testée. La table 3.2 montre les résultats obtenus en utilisant différents descripteurs de lieu. Le descripteur *Bag of Words* (BoW) est l'histogramme des mots visuels (clusters de descripteurs SIFT) calculé à l'intérieur de la façade candidate. VGG est le vecteur de taille 4096 obtenu en sortie du CNN VGG [SZ14] appliqué à la sous-image de la façade candidate, retaillée de manière à correspondre aux dimensions requises en entrée de ce réseau. SPP est le descripteur issu de la couche `conv5` du réseau SegNet modifié et SPP (siamois) est la sortie du réseau siamois obtenue pour ce même vecteur.

TABLE 3.2 – Statistiques concernant la reconnaissance de façade pour différents descripteurs de l'état de l'art et deux variantes de notre méthode.

	BoW	VGG	SPP (SegNet)	SPP (siamois)
Appariements corrects (%)	44.06	72.80	78.91	82.93
Temps (secondes)	0.29	2.19	0.07	0.05

Le faible nombre de points SIFT extraits sur les façades ainsi que la similarité de leurs descripteurs sur des structures répétées explique en grande partie la faible performance du descripteur BoW. En outre, ce descripteur n'intégrant pas d'information spatiale sur les points détectés, il permet uniquement de distinguer les façades en se basant sur les proportions de mots visuels représentés, ce qui est clairement insuffisant. La différence entre VGG et SPP est principalement imputable au réglage fin de SegNet modifié pour générer la segmentation. En effet, les architectures de ces deux réseaux sont comparables et SPP est essentiellement une approximation rapide à calculer d'un véritable descripteur CNN. En revanche, la couche `conv5` de notre réseau contient plus d'information sur les façades qu'un réseau VGG entraîné sur ImageNet. Enfin, on constate que l'apprentissage de métrique *via* un réseau siamois aide à la classification fine.
Nous évaluons à présent la méthode complète, intégrant la détection et la reconnaissance de façade. Des exemples de résultats sont montrés en figure 3.10. À la fin de l'étape de proposition de façade, nous obtenons 100 façades candidates. Comme l'étape d'appariement est basée sur la recherche du voisin le plus proche, nous pouvons difficilement l'appliquer sur chacun des

FIGURE 3.10 – Exemples de résultats de la méthode complète sur la base ZuBuD. Les polygones verts représentent les façades de la vérité terrain. Toutes les façades montrées dans cette figure ont été correctement reconnues.

candidats. En effet, en l'absence d'une prise en compte de la valeur de la distance obtenue avec le plus proche voisin (comparée à un seuil par exemple) chaque candidat trouvera toujours un voisin le plus proche parmi les images de référence, même si la façade sélectionnée n'existe pas dans l'image. Afin de ne pas biaiser les résultats par l'introduction d'un seuil, nécessairement différent pour chaque méthode, nous cherchons à reconnaître uniquement les façades effectivement visibles dans l'image. La classification de façade basée sur l'intérieur de la façade et sa région environnante permet d'éliminer la plupart des candidats. La validation croisée assure enfin que l'appariement est correct. Notre méthode obtient un taux de mauvaises détections de 16.14 % pour un taux de rappel de 71.13 % (SPP siamois) sur le jeu de test complet. Ces résultats sont meilleurs que ceux obtenus par le descripteur BoW et comparables à ceux obtenus avec le descripteur VGG, alors que la procédure de sélection de façade utilisée pour ces deux méthodes leur est favorable, puisque toujours basée sur le meilleur résultat pouvant être obtenu. Une fausse détection ne signifie pas nécessairement que la détection a échoué complètement. Elle peut aussi être expliquée par l'une ou l'autre des situations suivantes : la façade détectée est trop petite mais incluse dans une façade de la vérité terrain (image 1 en figure 3.11), la façade détectée contient plusieurs façades de la vérité terrain, regroupées en une seule (image 3 en figure 3.11), ou encore, la façade détectée n'a pas été détourée lors de la constitution de la vérité terrain (image 1 en figure 3.11). Cependant, nous avons identifié un cas d'échec récurrent qui se produit lorsque de petites façades détectées (petites dans l'image) contenant peu d'information photométrique sont appariées à une façade de référence quasi-homogène (image 1 en figure 3.11). L'appariement peut aussi échouer lorsque deux façades différentes de la base de référence sont très similaires, ce qui peut arriver pour des façades provenant d'un même bâtiment (image 4 en figure 3.11).

La méthode complète est enfin évaluée sur un plus petit jeu de données extrait de la partie Street

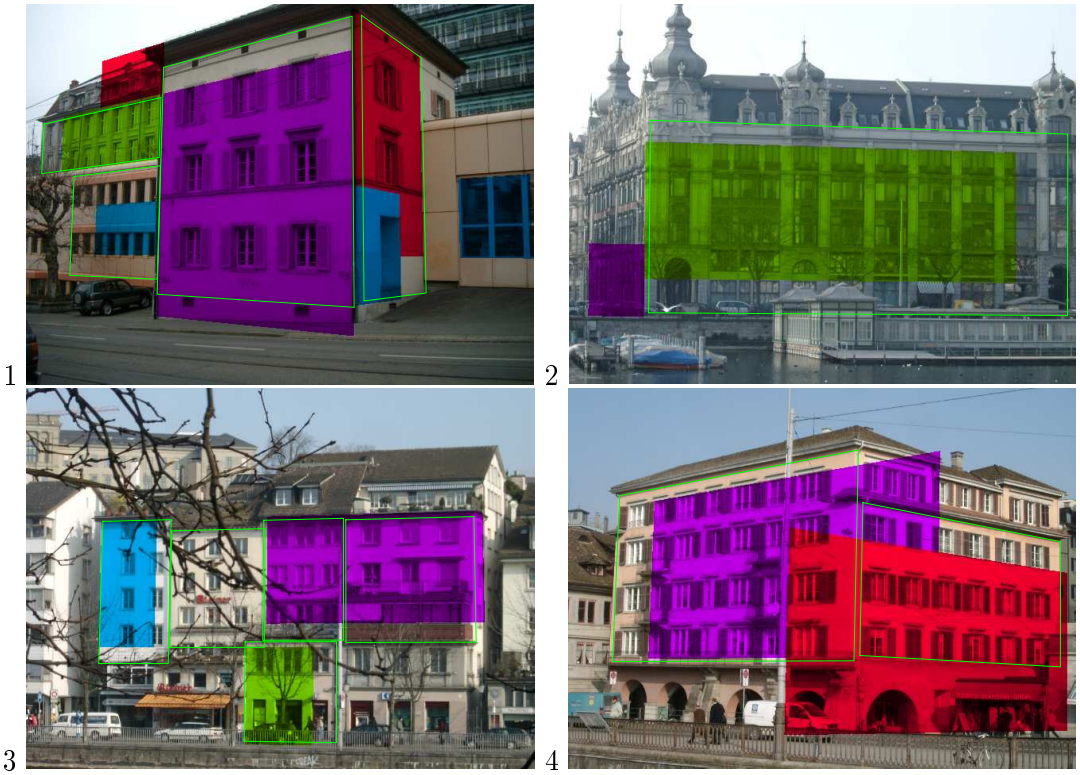


FIGURE 3.11 – Exemples d'échecs de la méthode avec différents cas de fausses détections (images 1, 2, 3) et un cas d'appariement de deux façades différentes avec la même façade de référence (image 4).

de la base *Cambridge Relocalisation Dataset*[6]. Ce jeu est composé de 80 images divisées en 20 classes. Cette situation est plus conforme à un cas d'utilisation de notre méthode où des données de géolocalisation obtenues par GPS permettent de restreindre le nombre d'images à considérer dans la base de référence. À la différence de ZuBuD, cette base contient très peu d'occultations, mais les changements de point de vue sont plus extrêmes. Les statistiques sont similaires à ce que nous obtenons sur ZuBuD, avec 73.38 % de taux de rappel et 19.34 % de mauvaises détections. Ces résultats montrent la robustesse de notre méthode aux forts changements de point de vue (figure 3.12).
Un autre intérêt de notre approche pour la reconnaissance de façade est sa rapidité. Comme nous utilisons des descripteurs SPP, nous avons besoin d'un seul passage dans le réseau SegNet modifié pour l'ensemble de la méthode. Nous exploitons les sorties de ce réseau, incluant les cartes de caractéristiques de la couche `conv5`, à différentes étapes de l'algorithme (échantillonage des rectangles, calcul des indices de façadité, calcul du descripteur, etc.). Cette méthode peut être vue comme méthode d'initialisation d'un algorithme plus précis de recalage 3D-2D, tel que ceux présentés dans [RD06a] et au chapitre 4 de ce mémoire. En tant que méthode d'initialisation, elle n'a pas besoin d'être exécutée pour chaque image du flux vidéo, mais uniquement au démarrage du suivi ou lorsque celui-ci échoue. La rapidité des calcul (typiquement, moins d'une demi-seconde par image soit environ une réinitialisation toutes les 10 images), est compatible avec des applications de RA.

3.5.3 Application à la RA

Une fois que des façades ont été détectées et identifiées dans une image, il est déjà possible d'y superposer des objets virtuels plans. Il suffit en effet d'appliquer à ces objets les homographies inverses de celles utilisées pour rectifier les images. Par exemple en figure 3.10, chaque région colorée représente une façade détectée et appariée avec succès. Ces régions pourraient être facile-

6. `http://mi.eng.cam.ac.uk/projects/relocalisation/#dataset`

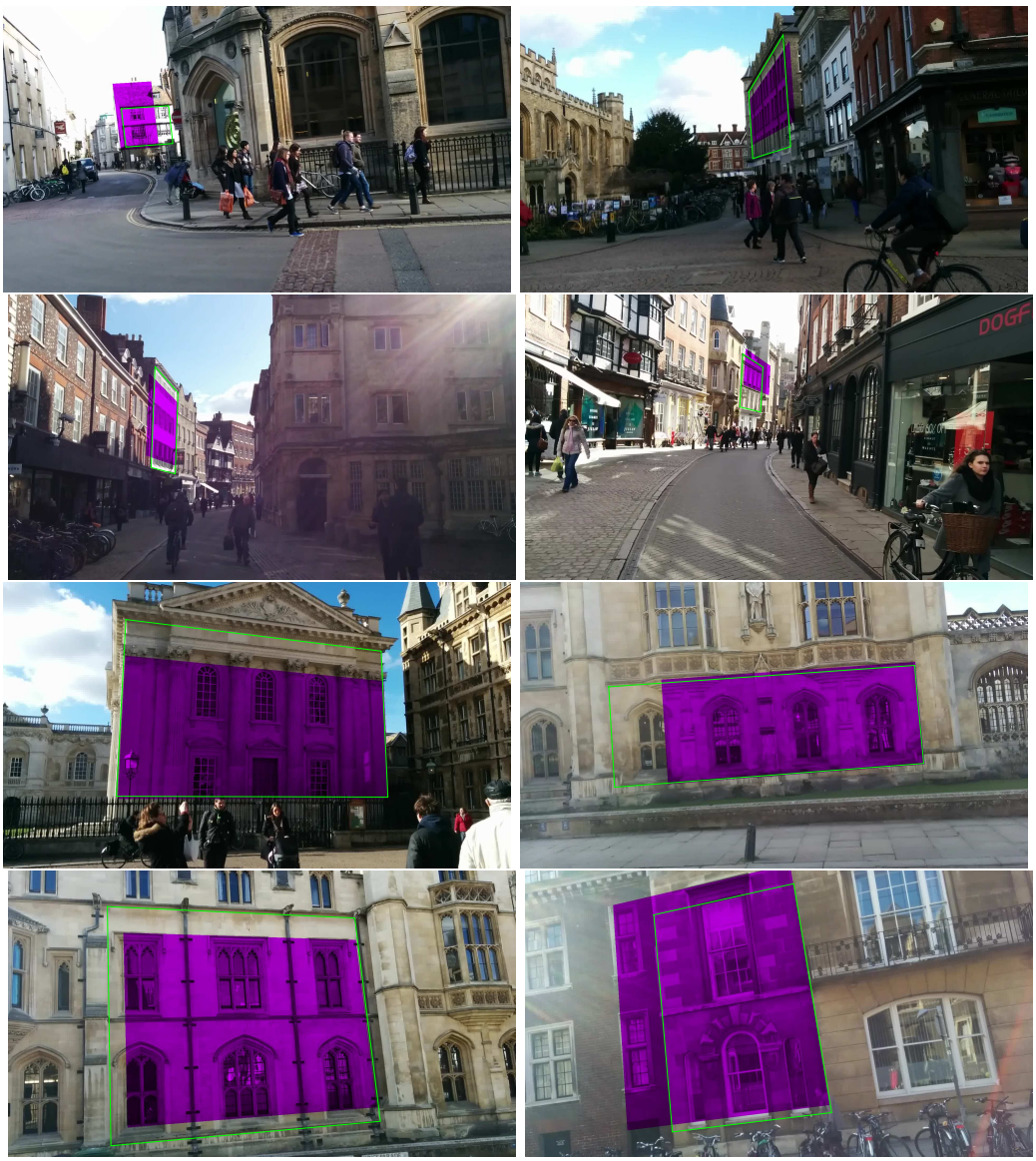


FIGURE 3.12 – Exemple de résultats obtenus avec la méthode de reconnaissance de façade complète sur le jeu de données de Cambridge. Toutes les façades montrées dans ces exemples ont été correctement reconnues.

ment remplacées par n'importe quelle information spécifique aux bâtiments concernés. De plus, si les façades de références sont associées à des informations géométriques géoréférencées (telles qu'un modèle 3D du bâtiment) la pose de la caméra peut être estimée par rapport à n'importe quelle façade reconnue (à partir de ses sommets, identifiés dans l'image par notre méthode), et donc dans le repère associé au géoréférencement. Pour illustrer cela, nous avons utilisé un modèle 3D public du Centre des Congrès de Nantes [7] et utilisé notre méthode pour tenter de reconnaître ce bâtiment parmi des images de GoogleStreetView le montrant sous différents angles de vue. Une façade de ce bâtiment a été ajouté à la base ZuBuD utilisée pour les expérimentations précédentes. Dans toutes les images de test la façade a été correctement détectée (apparaissant invariablement dans le top 10 de l'étape de proposition de façade) et reconnue. La figure 3.1 résume cette utilisation de la méthode dans les deux cas, incrustation d'un objet plan sur une façade en utilisant l'homographie inverse, ou projection d'un modèle 3D dans le cas où la façade est géoréférencée.

7. `http://www.3dwarehouse.sketchup.com`

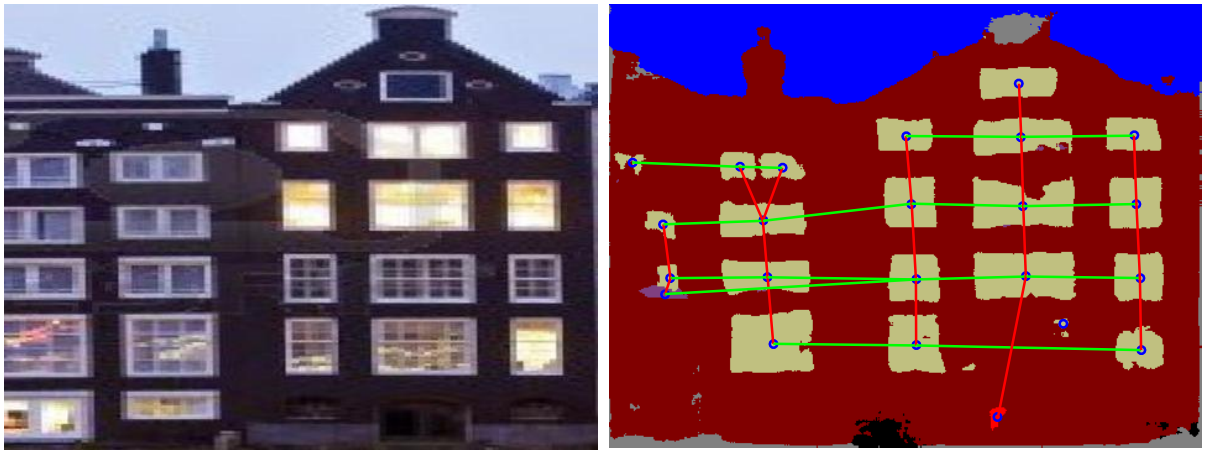
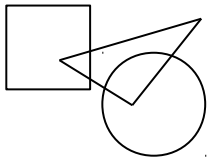


FIGURE 3.13 – A gauche une image de la base de données d'apprentissage. A droite son graphe de segmentation sémantique selon la méthode décrite en section 3.2.3 de la thèse d'Antoine Fond.

3.6 Conclusion et perspectives

Dans ce chapitre, nous avons décrit une méthode de détection et reconnaissance de façades qui repose sur une première étape de proposition de façades. Pour cette étape, nous avons adapté aux spécificités des milieux urbains trois indices issus de [ADF12]. Nous avons également proposé trois nouveaux indices (structure, symétrie et rectification) pertinents vis-à-vis de la structure sémantique et géométrique particulière des façades. Tous ces indices répondent à la contrainte de temps de calcul réduit attendue d'une méthode de proposition d'objets. 100 propositions triées selon un score combinant ces indices suffisent à détecter plus de 87% des façades de ZuBuD, ce qui dépasse largement les performances des méthodes de propositions d'objets génériques de l'état de l'art. S'ensuit une étape de classification forte basée sur des descripteurs CNN qui tiennent compte du contexte local de la façade. Ces mêmes descripteurs sont utilisés en entrée d'un réseau siamois pour reconnaitre les façades détectées parmi une base de références connues.
La méthode actuelle montre des bons résultats (environ 70% de rappel et 15% de fausses détections) sur des jeux de données urbains complexes. La rapidité de la méthode la rend compatible autant avec des applications de réalité augmentée simples qu'avec des applications plus complexes nécessitant une initialisation de calcul de pose de caméra. Les deux dernières étapes (classification et reconnaissance) pourraient toutefois bénéficier d'une description additionnelle plus robuste aux parties cachées, sous la forme d'un graphe reliant les différents éléments sémantiques d'une façade. Un tel graphe a été proposé à la fin de la thèse d'Antoine Fond pour calculer plus efficacement l'indice structurel utilisé pour la détection de façade (figure 3.13). Ce type de graphe pourrait être utilisé pour entraîner un classifieur par apprentissage profond adapté aux graphes [MBM+16]. Il pourrait également servir pour un test de compatibilité géométrique lors de la mise en correspondance avec la base de références.
Pour une utilisation effective de notre méthode, un problème de passage à l'échelle risque de se poser pour l'étape de reconnaissance. Si la base d'images ZuBuD contient déjà un millier d'images, ce nombre pourrait être beaucoup plus gros à l'échelle d'une ville. Bien-sûr, si des données GPS sont disponibles, il est possible de limiter la zone de recherche à un certain périmètre, et de réduire drastiquement le nombre d'images à considérer. En l'absence de cette connaissance, il reste à vérifier que les vecteurs en sortie du réseau siamois sont suffisamment espacés dans l'espace euclidien pour éviter les confusions lors de la recherche du plus proche voisin. Concernant les temps de calcul, il faut souligner que de nombreux algorithmes de recherche rapide du plus proche voisin ont été proposés ces dernières années (basés sur des méthodes de hashing, de quantification, sur des arbres, ...), et que ce thème de recherche est toujours très actif. Récemment, les auteurs de [HD16] ont par exemple proposé une nouvelle méthode de recherche basée sur un graphe, exploitant le fait que, pour de grandes bases de données, la dimension intrinsèque de la variété locale formée par les descripteurs est généralement bien inférieure à la dimension des descripteurs.



Ailleurs il y a du sang,
Ailleurs il y a du crime,
Des raisons qui n'en sont pas.

Vous nous avez dégagées
De ce qui n'était pas nous,
Qui vivait de quelque vie.

Et nous maintenant figées,
Sans colère, intempestives,
Nous collons à vos cornées.

Il faut vous en prendre à vous
Si vous souffrez de savoir
Que nous sommes quelque part.

Chapitre 4

Recalage par EM de labels sémantiques

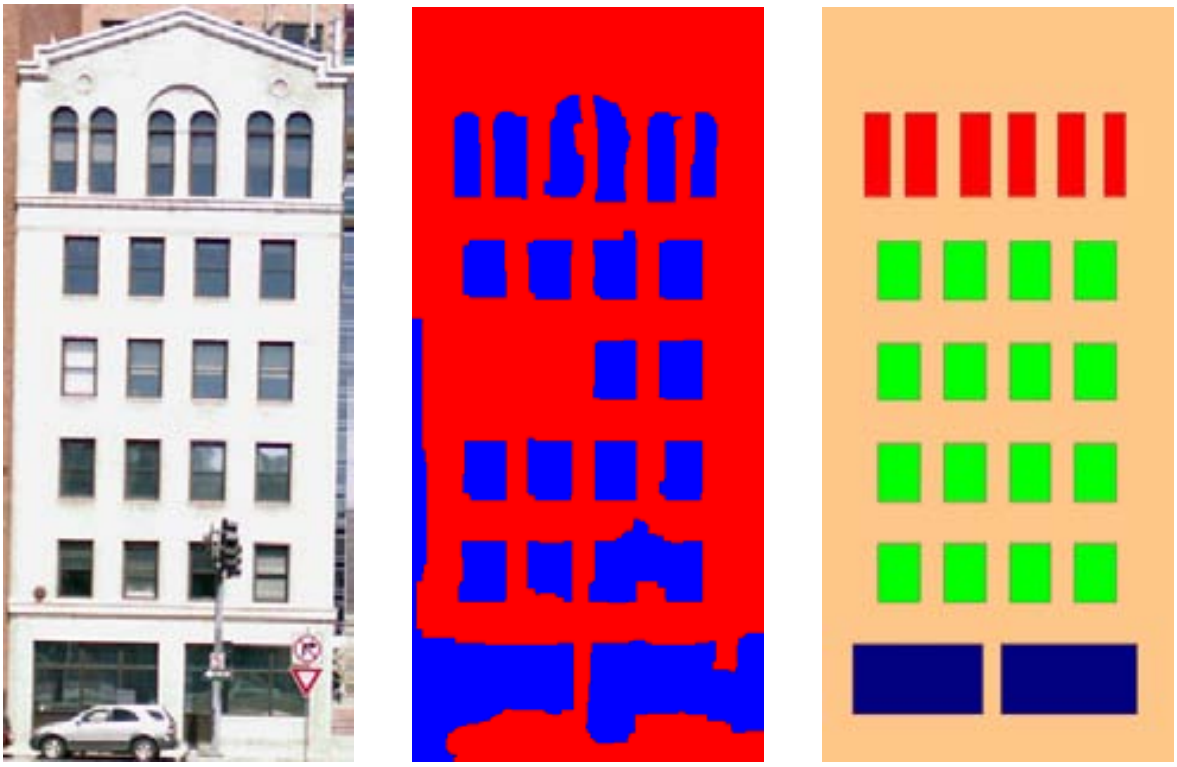


FIGURE 4.1 – À gauche une façade, au milieu sa segmentation sémantique inférée, à droite cette même segmentation sémantique régularisée par approximation de rang 1. Image issue de [YHQT12].

4.1 Introduction

La positionnement par synthèse (au sens large [RD06a, 19, APV+15, CWUF16]) a été en partie motivé par la disponibilité croissante de modèles 3D texturés à l'échelle nationale (IGN Géoportail) et planétaire (Google Street View/Maps, ...). Nous considérons dans ce chapitre le cas d'une seule façade de bâtiment, dont la vérité terrain d'une segmentation sémantique (découpage en régions correspondant aux fenêtres, portes etc.) est connue. Cette vérité terrain a été obtenue manuellement dans nos expérimentations, mais on peut envisager de l'obtenir automatiquement à partir de l'image de texture de la façade isolée de son contexte (figure 4.1(gauche)). Par exemple, les auteurs de [YHQT12] utilisent une classification de la texture en mur et non-mur (figure 4.1(milieu)), permettant de décrire la façade sous forme d'une matrice de 0 et de 1. Une méthode est alors proposée pour partitionner cette matrice en sous-blocs $M = M^0 + E$, tels que M^0 est de rang 1 (structure régulière) et E est une matrice résiduelle (erreurs de segmentation) dont la magnitude est faible au sens de la norme l^0 (figure 4.1(droite)). Une autre approche [TKS+13] considère un plus grand nombre de classes (mur, fenêtres, ...) et permet de décomposer une segmentation en une série de lois structurelles paramétriques appelée grammaire de forme (*shape grammar*), représentée par un arbre (figure 4.2). Une technique d'apprentissage par renforcement est utilisée pour obtenir l'arbre de segmentation optimal.

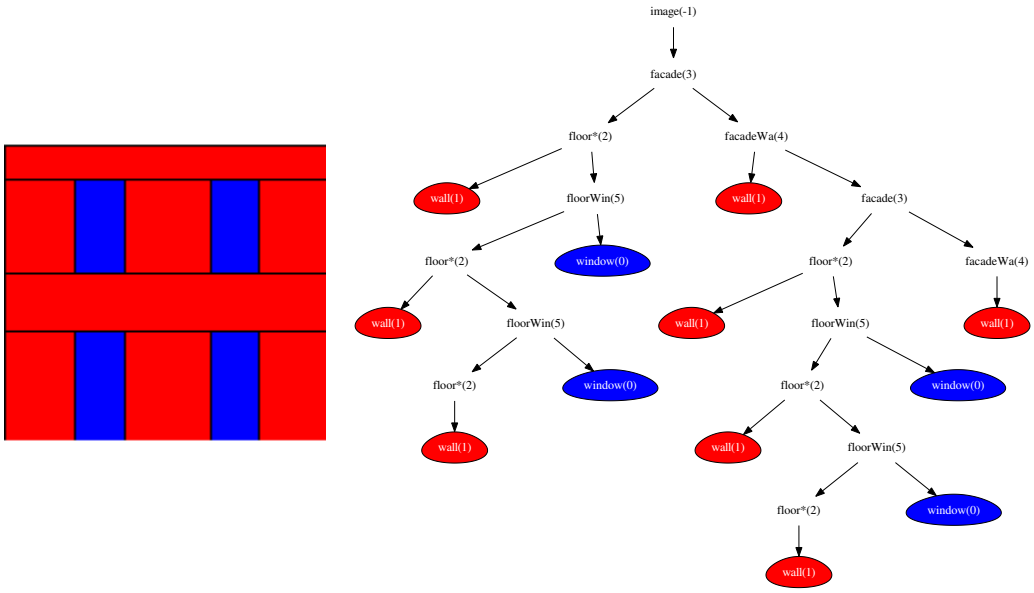


FIGURE 4.2 – Description d'une grammaire de façade représentée par un arbre. Image issue de [TKS+13].

Dans ce chapitre, nous montrons comment une façade représentée sous cette forme peut être recalée de manière robuste et précise dans une carte sémantique de l'image (obtenue par SegNet [BHC15]) partant d'une pose initiale. La pose initiale est ici obtenue à l'aide de la méthode décrite au chapitre 3. Bien que d'autres méthodes puissent être envisagées pour obtenir la pose initiale (GPS + magnétomètre, reconnaissance de lieux + points de fuite etc.), nous ne sommes pas en mesure de garantir que des résultats identiques à ceux présentés en section 4.5 puissent être obtenus avec ces variantes.

4.2 État de l'art et contributions

Selon notre décomposition du problème de positionnement global en milieu urbain, une fois la façade du bâtiment de l'image courante identifiée parmi les références du modèle, le second sous-problème vise à trouver la rotation $\mathbf{R}^\top$ et la position $-\mathbf{R}^\top\mathbf{t}$ de la caméra dans le repère de cette façade. Connaissant la matrice de calibration $\mathbf{K}$, ce problème de recalage peut être ramené à celui

du calcul de l'homographie $\mathbf{H}_w^i = \mathbf{K}\,(\mathbf{R}_1\mathbf{R}_2|\mathbf{t})$ qui transforme la façade (appartenant au plan d'équation $Z = 0$ dans le repère monde) de l'image de référence I_{ref} en la façade détectée dans l'image requête I, dite aussi image cible [32]. Si une homographie $\mathbf{H}_w^0$ est connue pour une image I_0, le problème revient alors à calculer l'homographie $\mathbf{H}_0^i$ qui transforme la façade de l'image I_0 en la façade de l'image I (on obtient alors $\mathbf{H}_w^i = \mathbf{H}_0^i\mathbf{H}_w^0$).
Dans le cadre du recalage par synthèse, l'image I_0 est l'image de synthèse de la façade, l'homographie $\mathbf{H}_w^0$ correspond à la pose approximative obtenue à l'aide de capteurs ou, dans notre cas, de la méthode décrite au chapitre 3, $\mathbf{H}_0^i$ est la correction à apporter à l'image de synthèse de la façade pour que celle-ci soit alignée avec l'image courante et enfin, $\mathbf{H}_w^i$ est la pose calculée pour l'image requête. Au chapitre 1, nous avons indiqué quels étaient les avantages du positionnement par synthèse par rapport à d'autres approches telles que l'utilisation de descripteurs locaux. Nous avons également précisé les limites des méthodes pouvant se réclamer de ce principe [RD06a, 19, APV+15, CWUF16].
En plus de ces méthodes, un grand nombre de procédures ont été proposées pour obtenir des mesures d'odométrie visuelle à partir du suivi de régions planes texturées. Ces méthodes ne sont pas concernées par l'obtention de l'homographie $\mathbf{H}_w^0$, qui est d'ailleurs souvent fixée à l'identité, mais uniquement par l'actualisation de l'homographie $\mathbf{H}_0^i$ tout au long de la séquence d'images. Cette actualisation étant récursive ($\mathbf{H}_0^i = \mathbf{H}_{i-1}^i \dots \mathbf{H}_1^2\mathbf{H}_0^1$), on observe en général un phénomène de dérive. Toutefois, rien n'interdit d'employer ces méthodes pour tenter de calculer l'homographie corrective $\mathbf{H}_0^i$ entre l'image de référence I_{ref} transformée par l'homographie approximative $\mathbf{H}_w^0$ et l'image cible I. Avec la réserve que la plupart de ces méthodes ont été évaluées dans le cadre d'un suivi de plan entre deux images consécutives d'une séquence vidéo, mais rarement dans le cadre d'un suivi de plan entre deux images rectifiées (ayant donc subi des changements de résolutions), acquises dans des conditions d'éclairage, de saison et de pose différentes. Nous avons vu par exemple au chapitre 1, que notre méthode de suivi de plan [32], basée sur un appariement de points d'intérêt entre deux images vidéo consécutives, n'est adaptée au recalage par synthèse que dans le cas où un flou de profondeur est appliqué de manière adéquate à l'image synthétisée. Une des contributions de ce chapitre est aussi d'évaluer l'apport possible d'autres méthodes de suivi de plan au recalage par synthèse.
On peut distinguer trois approches différentes pour calculer les homographies $\mathbf{H}_0^i$ par suivi de plan. Les approches denses appelés parfois *template-matching* cherchent à maximiser une mesure de similarité entre les images I et $I_{ref} \circ \mathbf{H}_0^i$. Les approches basées sur des primitives (*feature-based*) déduisent l'homographie par mise correspondance de points d'intérêt dans les deux images. Enfin il existe des méthodes qui estiment directement l'homographie à partir d'un modèle appris par régression.

4.2.1 Approches denses

Dans la première catégorie du recalage par méthodes denses, la mesure de similarité choisie est très importante. La première mesure utilisée a été la somme des différences de pixels au carrés entre les deux images (norme L_2) [LK+81]. L'optimisation se fait très rapidement par descente de gradient par l'algorithme de Gauss-Newton. Si la transformation était au début cantonnée à une simple translation 2D, les modèles géométriques ont ensuite été enrichis pour couvrir les transformations affines [HB98] et les homographies [BM01]. L'efficacité en temps de calcul de la minimisation a également été améliorée dans [BM04] par une approximation au second-ordre sans calculer le Hessien. La mesure de similarité par norme L_2 reste sensible aux changements d'illumination et aux occultations. Dans [JD02], l'image de référence est décomposée en une pyramide de sous-images recalées indépendamment vis-à-vis de l'image cible selon la norme L_2. La solution globale du recalage est cherchée récursivement dans l'espace des paramètres telle qu'elle maximise le nombre de sous-recalage. Si la décomposition permet de traiter efficacement les occultations, elle peut être sensible aux répétitions très fréquentes sur les façades.
Kim et al. [KF04] utilisent un M-estimateur pour une mesure de similarité plus robuste. L'information mutuelle entre les images est également une mesure de similarité moins sensible aux changements d'illumination et aux occultations [VWI97] et utilisée depuis longtemps pour le re-

calage multimodal en imagerie médicale [PMV03]. Si ces mesures augmentent significativement la complexité de l'optimisation, des progrès ont été apportés depuis qui permettent une résolution efficace [DM10] (figure 4.3). Néanmoins, toutes ces méthodes restent itératives et nécessitent une initialisation dont dépend fortement la convergence de l'algorithme vers la solution globale.

FIGURE 4.3 – Suivi par recalage d'image en utilisant l'information mutuelle comme mesure de similarité entre images [DM10]. L'homographie entre la référence (en bas) et l'image courante (en haut) est calculée à chaque pas de temps.

Dans un cas de recalage en translation seule, la solution globale peut rapidement être trouvée par décalage de phase dans le domaine fréquentiel. Cette méthode peut être généralisée à des similarités [RC96] et à des homographies [ZW05]. Cependant les zones de l'image courante qui ne correspondent pas à l'image de référence peuvent perturber la transformée de Fourier et faire ainsi échouer la méthode. Cela arrive régulièrement sur les images urbaines où un bâtiment peut être observé sous des échelles très différentes.

4.2.2 Approches basées sur des primitives

La seconde catégorie d'approches concerne les méthodes *feature-based*. Il s'agit d'extraire des points d'intérêt similaires dans les deux images et de les mettre en correspondance [32]. Les points SIFT [Low04] sont définis comme des maxima de l'espace d'échelle des différences de gaussiennes de l'image. Chaque point, associé à une échelle et une orientation caractéristique, est alors rendu invariant aux similarités. Les points sont mis en correspondance selon leur plus proche voisin vis-à-vis de leur descripteur calculé comme un histogramme de gradient orientés local. Couplé à la méthode d'estimation RANSAC cela permet de calculer l'homographie $\mathbf{H}_0^i$ entre les deux images, en étant robuste aux occultations et aux changements d'échelle d'observation importants.
Malgré ces atouts, le temps de calcul des descripteurs SIFT rend la méthode lente. La détection et la description de points d'intérêts robuste a été accélérée avec SURF [BTVG06b] en utilisant des réponses d'ondelettes de Haar calculées efficacement par images intégrales. ORB combine la rapidité d'extraction de FAST [RD06b] avec la rapidité de calcul des descripteurs BRIEF [CLSF10] pour offrir des performances similaires à SIFT en un temps très réduit. Si FAST utilisait déjà en partie de l'apprentissage automatique, l'émergence des réseaux de neurones convolutionels a conduit à LIFT [YTFLF16], une procédure calquée sur SIFT mais apprise de bout-en-bout (voir à ce sujet la section 1.2.1.1(p13), dans laquelle nous comparons LIFT et SIFT).

4.2.3 Régression par réseau de neurones convolutifs

Les performances des réseaux de neurones convolutifs profonds ont ouvert la voie à une nouvelle catégorie de méthodes de recalage visant à obtenir directement la transformation en sortie du réseau [DMR16]. Plutôt que de concaténer les deux images en entrée de réseau, Rocco et al. [RAS17] combinent les résultats d'un premier étage de description en une carte de corrélations à partir de laquelle un second réseau apprend à régresser la transformation. Ces approches par

régression sont directement reliées aux travaux de Kendall et al. [KGC15] présentés au chapitre 1, page 16, qui infèrent la pose à partir d'une seule image, le modèle de scène étant en quelque sorte inclus dans le réseau de neurones. Malheureusement, la robustesse intrinsèque de ces réseaux aux translations et aux petites déformations ainsi que la taille fixe des entrées limite la précision des transformation estimées.

4.2.4 Contributions

Dans les méthodes proposées par Arth et al. [APV+15] et Chu et al. [CWUF16], la segmentation sémantique est calculée une fois pour toutes, et les scores de classification les plus élevés de chaque pixel de l'image sont utilisés pour le recalage 3D-2D. Malheureusement, la segmentation peut comporter de nombreux pixels mal classifiés. L'originalité de notre approche est de remettre en cause la classification au cours d'un processus itératif de recalage, de type espérance-maximisation (EM). Il est important de noter que, comme pour [APV+15] et [CWUF16], notre méthode n'exige qu'un seul passage dans le FCN, au début du processus. La différence ici est que les scores de classification moins élevés que ceux des classes retenues définitivement par ces méthodes sont reconsidérés à chaque itération du recalage. Une porte peut par exemple être facilement classifiée en fenêtre en sortie du FCN, mais avec un score très peu supérieur à celui de la classe porte. Au fur et à mesure que la projection du modèle 3D associé à la vérité terrain de la segmentation, se rapproche de la projection attendue, notre méthode bayésienne est capable d'attribuer une plus grande vraisemblance à la classe porte qu'à la classe fenêtre, même si son score initial était moins élevé. Cet exemple, rencontré fréquemment en pratique, est illustré en figure 4.4.
Ainsi, si la segmentation sémantique est une donnée très intéressante pour guider le recalage, le recalage à son tour permet de guider la segmentation. Bien évidemment, comme la vérité terrain de la segmentation est connue, un recalage précis permet de retrouver cette vérité terrain dans l'image. Toutefois, la méthode que nous proposons est bien une méthode de recalage 3D-2D, pas une méthode de segmentation qui ne présenterait aucun intérêt en tant que telle, du fait que nous devons connaître la vérité terrain du résultat pour la ré-obtenir dans l'image.
Dans la suite du chapitre, nous commençons par indiquer comment l'algorithme EM est initialisé (section 4.3). Nous introduisons ensuite le modèle bayésien utilisé pour optimiser conjointement et itérativement la vraisemblance du recalage et de la segmentation (section 4.4). De nombreux résultats qualitatifs et quantitatifs sont enfin présentés en section 4.5. Nous montrons en particulier que notre approche surpasse d'autres approches correctives telles que ESM [BM04], la minimisation de l'information mutuelle [MHV+01, SSSC05] ou le classique SIFT + RANSAC [Low04], sur des jeux de données publics et un jeu de données illustrant l'alternance jour/nuit, que nous introduisons. Notre méthode s'avère en effet robuste aux occultations, aux motifs répétés, à l'alternance jour/nuit et aux artefacts de rectification, en raison principalement de la robustesse intrinsèque des FCN aux trois derniers facteurs, et de la robustesse de l'inférence bayésienne, en particulier aux occultations.

4.3 Initialisation

L'initialisation de la procédure EM (instant $t = t_0$) comporte quatre étapes :

1. les points de fuite ainsi que la distance focale de la caméra sont calculés en utilisant la version préalable [15] de la méthode présentée au chapitre 2. Le point principal est supposé au centre de l'image ;
2. l'image est rectifiée de manière à ce que les façades des bâtiments apparaissent en vue fronto-parallèle (plusieurs images peuvent être obtenues, plus précisément une image par point de fuite horizontal détecté) ;
3. les façades sont détectées (boites englobantes approximatives) et reconnues dans la ou les images(s) rectifiée(s), en utilisant la méthode présentée au chapitre 3.
4. la segmentation sémantique est calculée en utilisant SegNet [BHC15], et le recalage est initialisé à partir des boites englobantes approximatives.

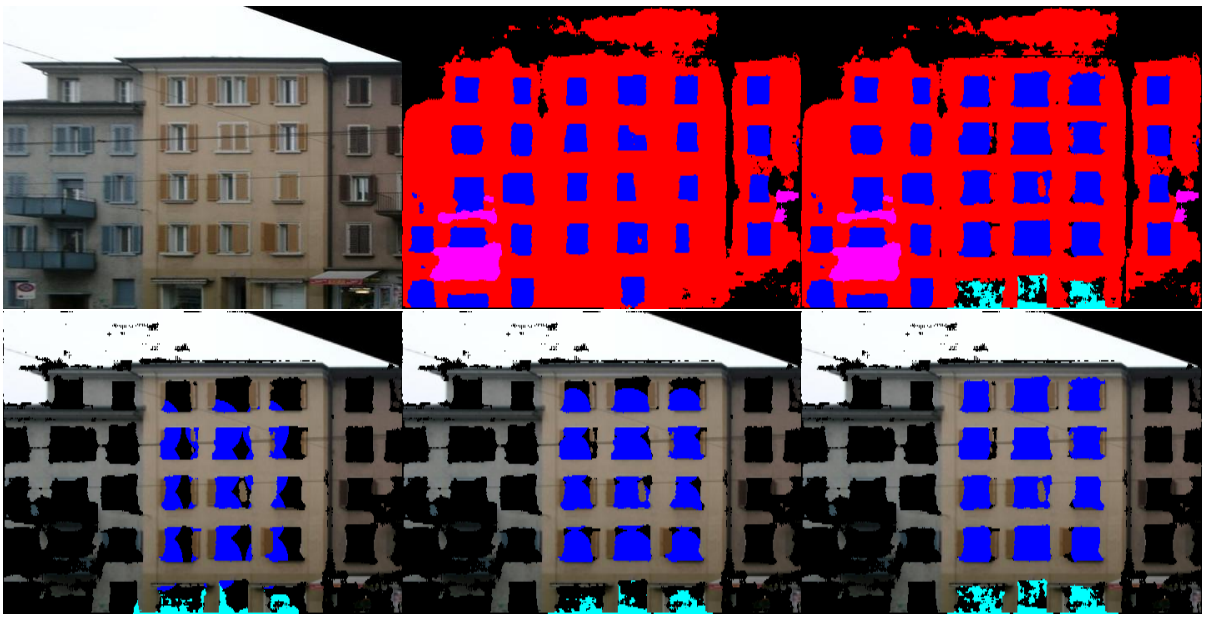


FIGURE 4.4 – Évolution de la segmentation sémantique au cours des trois premières itérations du recalage par EM. En haut (de gauche à droite) : l'image I donnée en entrée du FCN, dans laquelle on cherche à recaler l'immeuble orange ; la segmentation sémantique initiale $P(l_j|i, I)$; la segmentation sémantique finale, après recalage. En bas : les pixels associés aux portes du rez-de-chaussée, classifiés initialement en fenêtre ou en mur, sont progressivement correctement classifiés en même temps qu'ils permettent de guider le recalage.

L'image ayant été rectifiée en utilisant la connaissance de la rotation et des paramètres intrinsèques de la caméra, les paramètres de recalage restant à estimer (ou plutôt à affiner) sont un paramètre d'échelle s (le rapport d'aspect est préservé par la transformation homographique utilisée) et deux paramètres de translation (t_x, t_y). Une initialisation de ces paramètres est obtenue par simple résolution au sens des moindres carrés du système d'équations linéaires suivant :

$$\begin{pmatrix} x_{min} & 1 & 0 \\ y_{min} & 0 & 1 \\ x_{max} & 1 & 0 \\ y_{max} & 0 & 1 \end{pmatrix} \begin{pmatrix} s' \\ t'_x \\ t'_y \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ h \\ w \end{pmatrix}, \text{ avec } \begin{cases} s = \frac{1}{s'} \\ t_x = \frac{-t'_x}{s'} \\ t_y = \frac{-t'_y}{s'} \end{cases} \tag{4.1}$$

où (x_{min}, y_{min}) et (x_{max}, y_{max}) sont respectivement le coin en haut à gauche et le coin en bas à droite de la boite englobante détectée (approximative, cf. figure 4.5(en haut à gauche)) et (h, w) sont les dimensions de la façade reconnue (vérité terrain, cf. figure 4.5(en haut à droite)).
L'étape de détection de façade nécessite un passage de l'image rectifiée dans le réseau SegNet, et nous pourrions utiliser la carte sémantique obtenue à l'issue de cette étape pour initialiser la procédure EM. Malheureusement, l'inférence par SegNet est sensible à l'échelle (taille de la scène dans l'image), comme cela est illustré en figure 4.5(ligne du bas). Afin d'améliorer la précision de la segmentation initiale, nous imposons donc un deuxième passage dans le réseau, en donnant en entrée une image recadrée autour de la façade détectée (boite englobante élargie d'un facteur 1.4).

4.4 Recalage et segmentation sémantique simultanés

4.4.1 Modèle bayésien

Notre objectif est de recaler l'image de référence I_{ref} d'une façade dans une image cible I (*via* le calcul de la transformation T) tout en améliorant la qualité de la segmentation sémantique. Nous notons $L = \{l_j\}_{1 \leq j \leq K}$ les labels issus de la segmentation sémantique propres aux caractéristiques

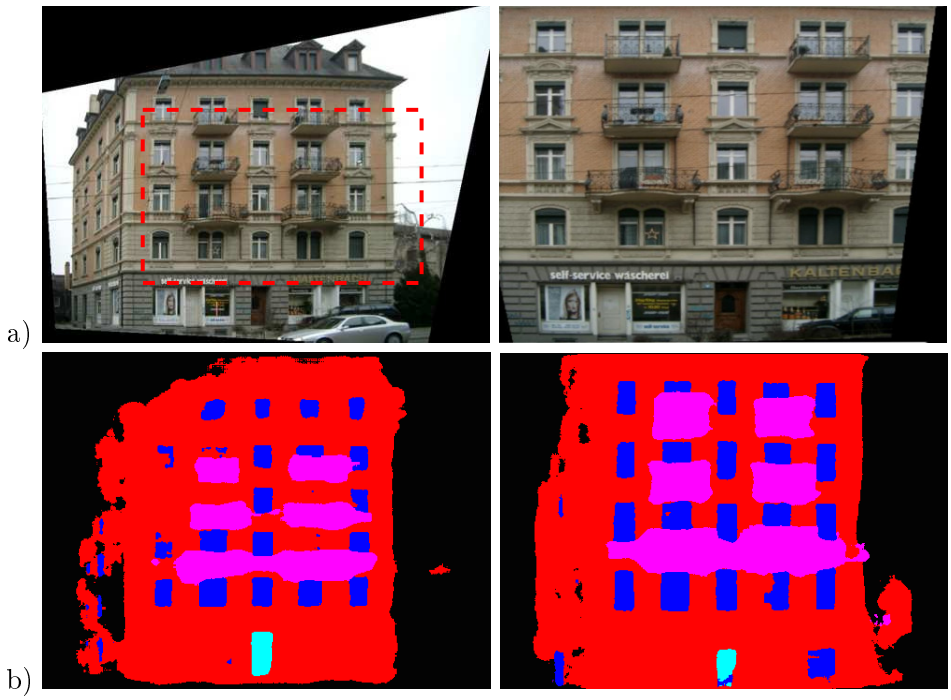


FIGURE 4.5 – Initialisation du recalage et de la segmentation : a) Recalage initial (rectangle hachuré en rouge dans l'image de gauche) de la façade de référence (image de droite). b) Segmentation initiale avant (gauche) et après le recadrage de l'image autour de la façade détectée.

architecturales d'une façade ("fenêtre", "porte", "balcon"). Les autres labels pouvant être obtenus à l'aide de SegNet ("façade","ciel", "route", "arrière-plan") ne sont pas considérés. L'image cible tout comme l'image de référence sont représentées par des ensembles de points 2D labélisés. Soit $X = \{X_i\}_{1 \leq i \leq N}$ un ensemble de N points $X_i = (x_i, y_i)$ de l'image cible I. Les coordonnées de ces points correspondent aux pixels i de l'image cible ayant une bonne probabilité de représenter un des labels considérés $P(l_j|i, I) \geq 0.01$ (figure 4.6). La probabilité $P(l_j|i, I)$ est le score de la dernière couche du CNN utilisé pour la segmentation sémantique.

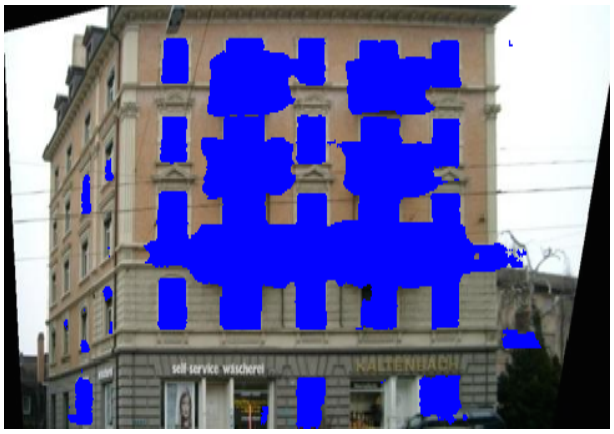


FIGURE 4.6 – Points X pour l'image cible I. Seuls les points susceptibles d'être caractéristiques d'un élément de façade ($P(l_j|i, I) \geq 0.01$) sont considérés.

L'ensemble des points X_{ref} de l'image de référence I_{ref} est représenté par des modèles de mélange de gaussiennes généralisées (GGMM) $\mathcal{N}_p$ de paramètres de forme p, en considérant un GGMM par label l_j : $\left(\pi_{k_j}, \mu_{k_j}, \Sigma_{k_j}\right)_{1 \leq k_j \leq m_j}$. Ces distributions sont bien adaptées aux éléments architecturaux d'une façade, la boule unité de la norme L_p $|M|^p_{p,\Sigma} = \frac{m_x^p}{\Sigma_{xx}} + \frac{m_y^p}{\Sigma_{yy}}$ étant proche d'un rectangle pour les valeurs élevées de p. Nous prenons $p = 4$, ce qui est raisonnable en terme de résolution du problème et permet déjà une bonne attache aux formes rectangulaires (figure

4.7(droite)). Le nombre de gaussiennes généralisées et leurs paramètres dépendent de l'image de référence I_{ref}, dont nous supposons connue la vérité terrain de la segmentation sémantique (figure 4.7(gauche)). Pour chaque label l_j de la vérité terrain, nous extrayons les composantes connexes sur lesquelles nous ajustons des gaussiennes généralisées (une gaussienne L_p par composante connexe). Cet ajustement est réalisé en deux temps. On cherche dans un premier temps à ajuster une gaussienne classique ($p = 2$) à chaque composante connexe. Le centre μ_{k_j} de cette gaussienne est positionné au barycentre des pixels de la région connexe. Comme l'image est rectifiée, les axes de chaque gaussienne sont alignés avec les axes de l'image. La covariance $\Sigma_{k_j} = \text{diag}\left(\sigma_x^{p/2}, \sigma_y^{p/2}\right)$ est initialisée à partir des variances des coordonnées horizontales et verticales des pixels de la région connexe (respectivement σ_x et σ_y). Dans un deuxième temps, le centre μ_{k_j} et la covariance Σ_{k_j} sont affinés en minimisant l'erreur entre la composante connexe et la gaussienne généralisée ($p = 4$) par un algorithme Gauss-Newton. Les poids du mélange $\left(\pi_{k_j}\right)_{1\leq j\leq K, 1\leq k_j\leq m_j}$ sont initialisés tel que π_{k_j} est le ratio entre le nombre de points de la composante connexe k_j et le nombre total de points dans l'image de référence. Ces poids sont ensuite normalisés de manière à ce que $\sum_{j,k_j} \pi_{k_j} = 1$.

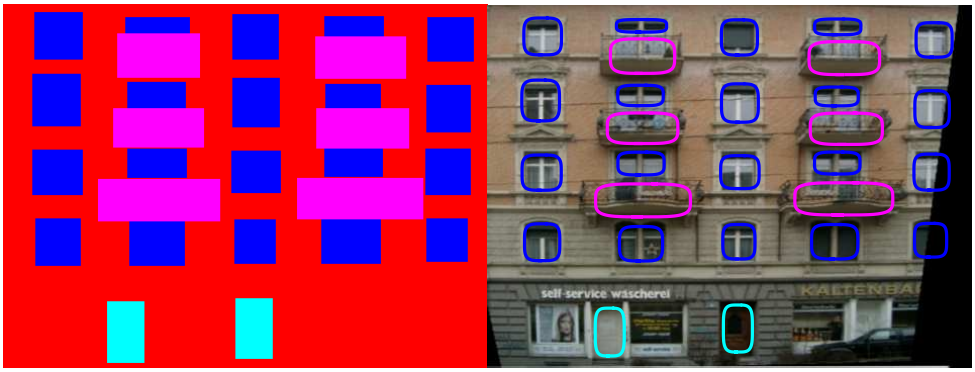


FIGURE 4.7 – Vérité terrain de la segmentation sémantique correspondant à l'image de référence I_{ref} (à gauche) et modèle de mélange de gaussiennes généralisées utilisé (à droite).

Le but est d'estimer la transformation géométrique $T(\Theta)$, de paramètres $\Theta = (t_x, t_y, s)$, permettant de recaler ces GGMM vers l'ensemble X des points observés dans l'image cible. En supplément, l'assignation d'un point X_i à un GGMM peut être vue comme une segmentation *a posteriori*. En supposant que les données observées X sont indépendantes, la vraisemblance de la distribution *a posteriori* est maximisée pour trouver Θ :

$$\begin{aligned}\Theta^\star &= \underset{\Theta}{\text{argmax}} \ln P(X|\Theta, I)P(\Theta) \\ &= \underset{\Theta}{\text{argmax}} \sum_{i=1}^{N} \ln P(X_i|\Theta, I) + \ln P(\Theta)\end{aligned} \tag{4.2}$$

La formule des probabilités totales permet d'exprimer $P(X_i|\Theta, I)$ de la manière suivante :

$$\begin{aligned}P(X_i|\Theta, I) = &\sum_{j=1}^{K} P(X_i|l_j, \Theta, I)P(l_j|i, \Theta, I) \\ &+ P(X_i|o, \Theta, I)P(o|i, \Theta, I),\end{aligned} \tag{4.3}$$

où :

— $P(X_i|l_j,\Theta,I)$ est donnée par la transformée du GGMM associé au label l_j :

$$\begin{aligned} P(X_i|l_j,\Theta,I) &= \sum_{k_j=1}^{m_j} \pi_{k_j} \mathcal{N}_p\left(X_i|T\mu_{k_j}, s^p\Sigma_{k_j}\right) \\ &= \sum_{k_j=1}^{m_j} \pi_{k_j} \frac{\exp\left(-|X_i - T\mu_{k_j}|^p_{p,s^p\Sigma_{k_j}}\right)}{4/p^2\Gamma(1/p)^2|s^p\Sigma_{k_j}|} \end{aligned} \tag{4.4}$$

— $P(l_j|i,\Theta,I)$ est la probabilité de segmentation *a priori*. Grâce au recadrage et à l'invariance du CNN aux faibles translations, l'inférence de la segmentation sémantique est relativement stable. Nous supposons donc que $P(l_j|i,\Theta,I) = P(l_j|i,\Theta^{(t_0)},I)$.
— $P(X_i|o,\Theta,I) = P(X_i|o,I)$ est la probabilité que X_i soit un point anormal (*outlier*), modélisée par une distribution spatiale uniforme $P(X_i|o,I) = \frac{1}{HW}$ avec H,W les dimensions de l'image cible I,
— $\alpha = P(o|i,\Theta,I)$ est le taux d'anomalies attendu, initialisé à $\alpha = 0.25\left(1 - \frac{s^2hw}{HW}\right)$ avec h,w les dimensions de l'image de référence I_{ref}.

Afin d'être plus robuste à la présence d'objets ou de piétons occultant la façade, nous laissons varier les poids de la mixture durant l'inférence. Ces poids sont donc ajoutés aux paramètres de l'inférence, $\Theta = \left(\{\pi_{k_j}\}_{1\leq j\leq K, 1\leq k_j\leq m_j}, \alpha, t_x, t_y, s\right)$, ce qui ne change en rien l'équation 4.2. Nous ne considérons aucun *prior* pour les paramètres (t_x,t_y,s). En revanche, afin d'éviter que les poids de la mixture ne dérivent, nous supposons que ceux-ci suivent une distribution de Dirichlet :

$$P(\Theta) = \mathcal{D}ir\left(\pi_{k_j}|\alpha_{k_j}\right)_{1\leq j\leq K, 1\leq k_j\leq m_j} \propto \prod_{j,k_j} \pi_{k_j}^{\alpha_{k_j}-1} \tag{4.5}$$

Gauvain et al. [GL94] ont montré que la distribution de Dirichlet est commode pour les mélanges de distributions car elle permet d'établir des formules analytiques aux solutions des équations de l'EM. Les paramètres $\left(\alpha_{k_j}\right)_{1\leq j\leq K, 1\leq k_j\leq m_j}$ de la distribution Dirichlet sont fixés aux mêmes valeurs que les poids initiaux du mélange $\alpha_{k_j} = \pi_{k_j}^{(t_0)}$.

4.4.2 Résolution par espérance-maximisation

Le problème de Maximisation *A Posteriori* (MAP) exprimé par l'équation (4.2) peut être résolu dans un cadre EM. On définit les variables latentes

$$Z = \left\{z_{i,j,k_j} \in \{0,1\}, z_{i,o} \in \{0,1\}\right\}_{1\leq i\leq N, 1\leq j\leq K, 1\leq k_j\leq m_j}$$

telles que z_{i,j,k_j} assigne un point X_i à une gaussienne $L_p\left(T\mu_{k_j}, s^p\Sigma_{k_j}\right)$ de l'étiquette l_j et $z_{i,o}$ assigne X_i à la classe d'anomalie supplémentaire o. L'algorithme EM cherche à trouver la solution itérativement en alternant entre le calcul de l'espérance (par rapport à Z) de la log-vraisemblance complétée $Q(\Theta|\Theta^{(t)})$ conditionnellement à X et aux paramètres courants $\Theta^{(t)}$, et la recherche des paramètres Θ qui maximisent cette quantité :

$$\begin{aligned} Q\left(\Theta|\Theta^{(t)}\right) &= \mathbb{E}_{Z|X,\Theta^{(t)}} \ln P(X,Z|\Theta) \\ &= \sum_Z P(Z|X,\Theta^{(t)}) \ln P(X,Z|\Theta) \\ &= \sum_{i,j}\sum_{k_j} \beta_{i,j,k_j}\left(\ln\pi_{k_j} + \ln P(l_j|i,\Theta,I)\right) \\ &+ \sum_{i,j}\sum_{k_j} \beta_{i,j,k_j} \ln \mathcal{N}_p\left(X_i|T\mu_{k_j}, s^p\Sigma_{k_j}\right) \\ &+ \sum_i \gamma_i \ln\frac{\alpha}{HW} \end{aligned} \tag{4.6}$$

avec $\beta_{i,j,k_j} = \mathbb{E}\left(z_{i,j,k_j}|X,\Theta^{(t)}\right)$ et $\gamma_i = \mathbb{E}\left(z_{i,o}|X,\Theta^{(t)}\right)$.

Ainsi l'algorithme EM itère entre ces deux étapes :

— **E-Step :** calcul de β_{i,j,k_j} et γ_i

— **M-Step :** calcul de $\Theta^{(t+1)} = \text{argmax}_\Theta\, Q\left(\Theta|\Theta^{(t)}\right) + \ln P(\Theta)$

L'étape **E-Step** peut être vue comme le calcul de la probabilité d'assignation de chaque point de donnée observé X_i à une Gaussienne $L_p\left(T\mu_{k_j}, s^p\Sigma_{k_j}\right)$ de l'étiquette l_j connaissant les paramètres $\Theta^{(t)} = \left(\{\pi_{k_j}\}^{(t)}_{1\leq j\leq K\,,1\leq k_j\leq m_j}, \alpha^{(t)}, t_x^{(t)}, t_y^{(t)}, s^{(t)}\right)$. En utilisant la règle de Bayes et en notant $\lambda = \frac{\alpha}{HW}$, on peut écrire :

$$\begin{aligned}\beta_{i,j,k_j} &= \mathbb{E}\left(z_{i,j,k_j}|X,\Theta^{(t)}\right)\\ &= \frac{\pi_{k_j}\mathcal{N}_p\left(X_i|T\mu_{k_j}, s^p\Sigma_{k_j}\right)P(l_j|i,\Theta,I)}{\sum_{j',k'}\pi_{k'_{j'}}\mathcal{N}_p\left(X_i|T\mu_{k'_{j'}}, s^p\Sigma_{k'_{j'}}\right)P(l_{j'}|i,\Theta,I) + \lambda}\end{aligned} \tag{4.7}$$

$$\begin{aligned}\gamma_i &= \mathbb{E}\left(z_{i,o}|X,\Theta^{(t)}\right)\\ &= \frac{\lambda}{\sum_{j',k'}\pi_{k'_{j'}}\mathcal{N}_p\left(X_i|T\mu_{k'_{j'}}, s^p\Sigma_{k'_{j'}}\right)P(l_{j'}|i,\Theta,I) + \lambda}\end{aligned} \tag{4.8}$$

Dans l'étape **M-Step** nous visons à maximiser $R = Q\left(\Theta|\Theta^{(t)}\right) + \ln P(\Theta)$ connaissant les assignations $\beta_{i,j,k}$ et γ_i. En développant les expressions des distributions des équations 4.4 et 4.5 et en ignorant les termes constants, R peut être réécrit $\tilde{R}$:

$$\begin{aligned}\tilde{R} = &-\sum_{i,j,k_j}\frac{\beta_{i,j,k_j}}{2}\left(\ln|s^p\Sigma_{j,k_j}| + |X_i - T\mu_{k_j}|^p_{p,s^p\Sigma_{j,k_j}}\right)\\ &+\sum_{i,j,k_j}\beta_{i,j,k_j}\ln\pi_{k_j} + \sum_i\gamma_i\ln\lambda + \sum_{j,k_j}\left(\alpha_{k_j}-1\right)\ln\pi_{k_j}\end{aligned} \tag{4.9}$$

Des dérivées partielles $\frac{\partial\tilde{R}}{\partial t_x} = \frac{\partial\tilde{R}}{\partial t_y} = \frac{\partial\tilde{R}}{\partial s} = 0$ on peut dériver un système polynomial qui ne peut être résolu de manière analytique pour $p = 4$. Notre stratégie de résolution est similaire à l'approche utilisée pour l'initialisation des paramètres du mélange sur la sémantique de référence. Premièrement, nous résolvons analytiquement le système polynomial avec $p = 2$ qui correspond à un mélange de Gaussiennes classique (annexe B). Puis nous raffinons le résultat en minimisant $J = \frac{\partial\tilde{R}}{\partial t_x}^2 + \frac{\partial\tilde{R}}{\partial t_y}^2 + \frac{\partial\tilde{R}}{\partial s}^2$ pour $p = 4$ par descente de gradient. Comme J est polynomiale, le gradient comme le Hessien peuvent être calculés rapidement en utilisant leur expression polynomiale dans l'algorithme de Gauss-Newton (les détails de ces calculs peuvent être trouvés dans l'annexe B de la thèse d'Antoine Fond [Fon18]). La convergence est atteinte après quelques itérations et on met à jour les paramètres de la transformation $\left(t_x^{(t+1)}, t_y^{(t+1)}, s^{(t+1)}\right)$. La mise à jour des poids π_{k_j} et le taux d'anomalies α suivent les formules établies par [GL94] :

$$\pi_{k_j}^{(t+1)} = \frac{\sum_i\beta_{i,j,k_j} + \alpha_{k_j} - 1}{\sum_{i,k'_{j'}}\beta_{i,j,k_{j'}} + \sum_{k'_{j'}}\left(\alpha_{k'_{j'}} - 1\right)} \tag{4.10}$$

$$\alpha^{(t+1)} = \frac{\sum_i\gamma_i}{\sum_{i,k'_{j'}}\beta_{i,j,k_{j'}} + \sum_{k'_{j'}}\left(\alpha_{k'_{j'}} - 1\right)} \tag{4.11}$$

4.5 Resultats expérimentaux

4.5.1 Implémentation et efficacité

La représentation compacte d'une façade utilisée par notre méthode (paramètres des GGMM) rend celle-ci particulièrement efficace. Le nombre de gaussiennes généralisées utilisées est de l'ordre du nombre de fenêtre sur une façade (typiquement entre 2 et 30). Si l'on suppose en outre que l'image est pleine de façades adjacentes et que l'espace vide entre les fenêtres est aussi grand que les fenêtres elles-mêmes, nous pouvons approximer le nombre de points par $N \approx 0.25HW$. Dans nos données de test, cette approximation est à peu près correcte, avec une moyenne de $\hat{N} = 31000$. En pratique, le recalage ne nécessite pas que des points soient échantillonnés sur chaque pixel de l'image. Dans notre implémentation nous utilisons une approche multi-résolution à deux niveaux. L'algorithme EM est d'abord exécuté sur une version sous-échantillonnée de l'ensemble de points X jusqu'à convergence ($|\Theta^{(t+1)} - \Theta^{(t)}| \leq \epsilon$), puis ré-exécuté sur l'ensemble de points complet X à partir de la dernière estimée $\Theta^{(t)}$.
La complexité d'une itération t de l'algorithme EM est en $O\left(NK \max_j m_j\right)$ et une parallélisation de l'étape E-step est tout à fait possible dans la mesure où les calculs de β_{i,j,k_j} sont indépendants. La complexité réduite de notre algorithme est aussi une conséquence de la résolution partielle de l'étape M-step en *closed-form* suivie de quelques itérations de Gauss-Newton négligeables en terme de temps de calcul. Le code de notre implémentation est écrit en Matlab sauf la partie EM qui est compilée à partir de code C. Le temps de calcul moyen d'une itération t est de 0.023 sec. sur un CPU I7-3520M. Le nombre d'étapes requis pour la convergence de l'algorithme EM dépend beaucoup de l'initialisation. Dans nos données de test, seulement 6 itérations sont nécessaires pour converger au niveau sous-échantillonné, et 2 itérations supplémentaires sont requises pour converger au niveau plus dense. Notre approche d'optimisation M-Step est ainsi plus rapide et plus précise sur ce problème que les méthodes basées sur une continuité homotopique, avec un temps de calcul moyen pour l'ensemble de la procédure EM de 0.121 sec.
En pratique, afin d'éviter des problèmes de convergence vers un maximum local, nous utilisons plusieurs initialisations. Nous appliquons notre méthode, non seulement sur la façade détectée, mais aussi sur les 20 premières détections rendues par la méthode présentée au chapitre 3, qui recouvrent partiellement la façade détectée. La solution finalement retenue est celle qui aboutit à la vraisamblance (valeur de R) la plus élevée.

4.5.2 Évaluation de la méthode

4.5.2.1 Protocole expérimental

Notre méthode a été évaluée sur trois jeux de données. Le premier est VarCity 3D [1]. Ce jeu de données contient 401 images de bâtiments acquises par un humain se déplaçant le long d'une seule rue. Ces images sont labellisées du point de vue sémantique et une reconstruction 3D de la scène, ainsi que les paramètres extrinsèques et intrinsèques de la caméra sont connus. Les points de vue d'observation sont à peu près fronto-parallèles aux façades et les façades couvrent une grande partie des images. Les changements d'échelle par rapport aux images de référence sont donc généralement faibles, mais les composantes de la translation peuvent être élevées, impliquant fréquemment une sortie partielle de la façade du champ de vision de la caméra.
Le second jeu de données est constitué des 100 premiers bâtiments de la Zurich Buildings Database (ZuBuD), chacun étant photographié selon 5 points de vue différents. Parmi ces scènes nous conservons uniquement celles qui ont pu être correctement reconstruites par SFM [2]. La diversité des points de vue sur ce jeu de données permet d'évaluer la robustesse de la méthode à de plus grands changements d'échelle, ainsi qu'à la présence d'occultations.
Le troisième jeu de données, que nous avons appelé NancyLights, vise à montrer la robustesse de la méthode aux changements d'illumination. Il contient 56 images d'une même façade, prises

1. https ://varcity.ethz.ch/3dchallenge
2. http ://ccwu.me/vsfm

depuis le même point de vue à des moments différents de la journée, allant du jour à la nuit. Pour chaque bâtiment dans chacun des trois jeux de données, nous sélectionnons la façade de référence parmi les images où celle-ci apparaît en vue la plus fronto-parallèle possible et avec le moins d'occultations possible. La référence est segmentée manuellement en trois labels sémantiques "fenêtre", "porte" et balcon (figure 4.7). Afin d'évaluer la précision de la méthode, la vérité terrain des frontières de la façade est transférée dans toutes les images où cette façade est visible en utilisant l'information géométrique obtenue par SFM.

4.5.2.2 Résultats

Notre méthode a été comparée à d'autres méthodes de recalage basées sur l'image ou sur des primitives détectées dans l'image. Dans la première catégorie de méthode, nous comparons le résultat du recalage à la détection brute obtenue en sortie de la méthode présentée au chapitre 3, au résultat obtenu par minimisation de la différence d'intensité (entre l'image cible et l'image recalée) utilisant la norme L_2 et une descente de gradient [BM04], à la maximisation de l'information mutuelle [MHV+01, SSSC05], et à la corrélation de phase [RC96]. Nous utilisons dans chacun des cas les mêmes paramètres de transformation initiaux que ceux utilisés avec notre méthode. Dans la deuxième catégorie de méthode, nous extrayons des descripteurs SIFT dans l'image cible (rectifiée) et dans l'image de référence. L'algorithme RANSAC est utilisé pour calculer la transformation géométrique entre les deux images, à l'aide de tirages aléatoires de deux pairs de descripteurs appariés selon le critère de Lowe [Low04]. La comparaison est effectuée en calculant l'histogramme cumulé normalisé des erreurs sur la translation et l'échelle (figure 4.8). Pour ZuBuD et VarCity, le modèle acquis par SFM permet de montrer également l'impact de l'erreur de recalage 2D sur la translation 3D de la pose de la caméra (table 4.1).

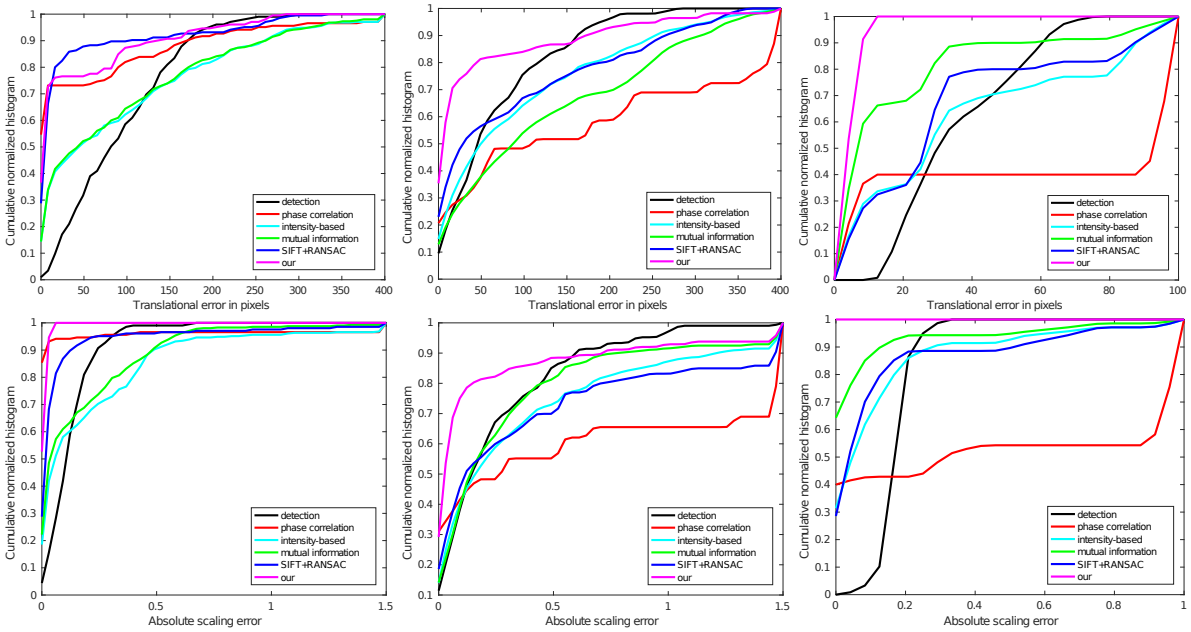


FIGURE 4.8 – Erreurs de recalage obtenues sur VarCity3D (gauche), ZuBuD (milieu) et Nancy-Lights (droite).

	SIFT+RANSAC	phase correlation	intensity-based	mutual information	ours
VarCity 3D	0.04	0.02	0.37	0.35	0.03
ZuBuD	0.22	0.67	0.33	0.44	0.12

TABLE 4.1 – Erreurs médianes obtenues pour la translation 3D de la caméra (distance entre la position attendue et la position obtenue, divisée par la distance de la caméra à la façade).

Les bons résultats obtenus par notre méthode sur VarCity 3D montrent que celle-ci peut trai-

ter correctement des cas de translations importantes grâce au support infini des Gaussiennes L_p. Même quand ce phénomène coïncide avec la présence de motifs répétés, les initialisations multiples qui exploitent ces répétitions et symétries ainsi que la régularisation du MAP aident à estimer un recalage correct. Au contraire, ces situations sont le défaut majeur des méthodes d'optimisation basées sur l'image qui sont facilement attirées vers des optima locaux (figure 4.9(ligne du haut)). Malgré tout, notre méthode peut également échouer dans quelques-uns de ces cas lorsqu'un manque de composantes architecturales discriminantes, comme une porte, lui fait préférer un mauvais alignement d'étage ou de fenêtres (figure 4.10(gauche)). SIFT résout presque tous ces cas précis en profitant d'autres éléments visuels de la façade.

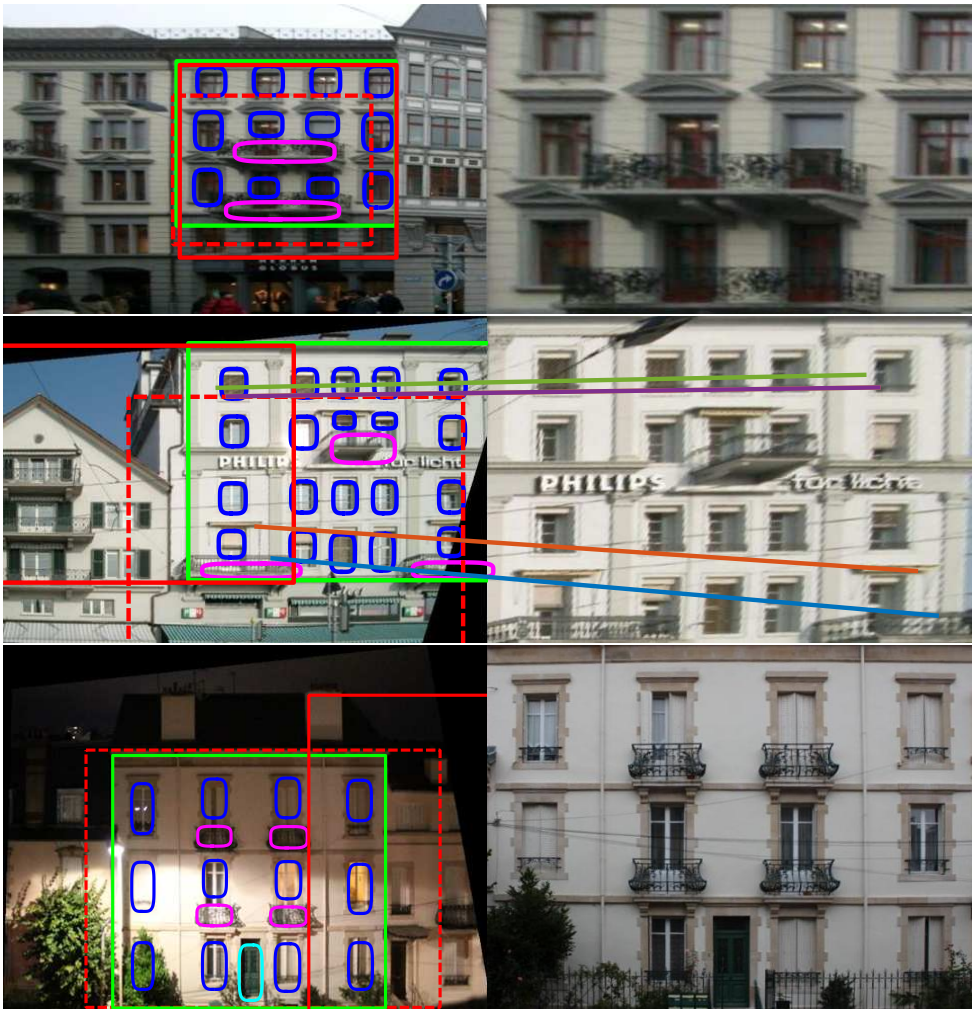


FIGURE 4.9 – Exemples de résultats où d'autres méthodes échouent (rectangles rouges) quand la notre réussit (rectangles verts). Le recalage initial est représenté en trait hachuré, les recalages finaux en trait plein. En-haut : le recalage basé sur l'intensité tombe dans un minimum local. Au milieu : le recalage par SIFT/RANSAC échoue en raison du caractère symétrique de la façade. En-bas : le grand changement d'illumination fait échouer la méthode par corrélation de phase.

Le jeu de tests ZuBuD met l'accent sur d'autres situations complexes, la diversité de points de vues générant des changements d'apparence dans les images rectifiées, notamment en échelle. La rectification peut par ailleurs induire des artefacts de très faible résolution. Dans ces conditions, peu de descripteurs SIFT sont extraits et ils se ressemblent tous, ce qui peut être source de mauvais recalage (figure 4.9(ligne du milieu)). Comme le recalage est borné à la seule façade, il peut échouer là où un algorithme SFM classique peut exploiter des éléments visuels du contexte. D'un autre côté, notre approche peut bénéficier d'une détection initiale relativement proche de la solution. Les occultations sont une autre conséquence de la diversité de points de vue sur ce jeu de tests. La mise à jour des poids du mélange pendant l'EM permet à notre méthode d'être

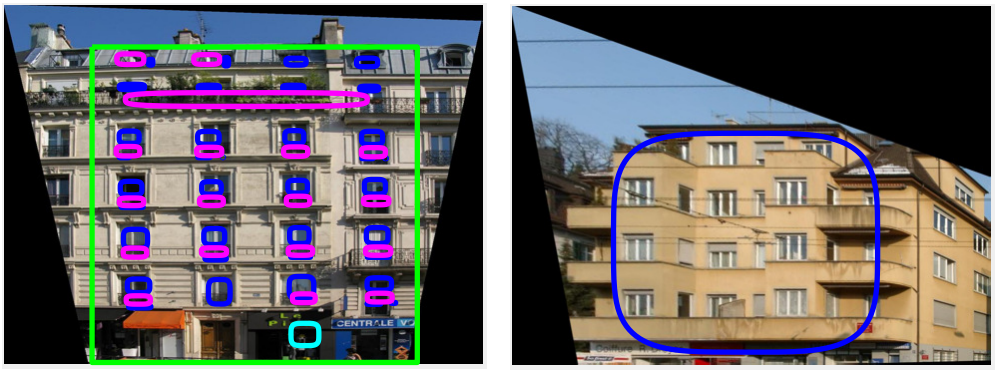


FIGURE 4.10 – Exemple d'erreurs de recalage de notre méthode. À gauche le résultat de notre méthode est décalé d'une colonne de fenêtre, ne pouvant s'aligner avec les fenêtres de droite partiellement visibles. A droite la présence du seul élément sémantique dense "fenêtre" conjugué à une géométrie non conforme à l'hypothèse de façade plane a conduit notre méthode à finir dans un minimum local.

robuste à celles-ci ainsi qu'aux parties en dehors du champ de vision de la caméra car la valeur de π_{k_j} peut décroître si un élément n'est pas visible (figure 4.11). Opérant comme critère de régularisation, la distribution *a priori* de Dirichlet sur les poids du mélange évite également la complète ignorance des données en gardant les poids proches de leur valeur initiale α_{k_j} (figure 4.12).

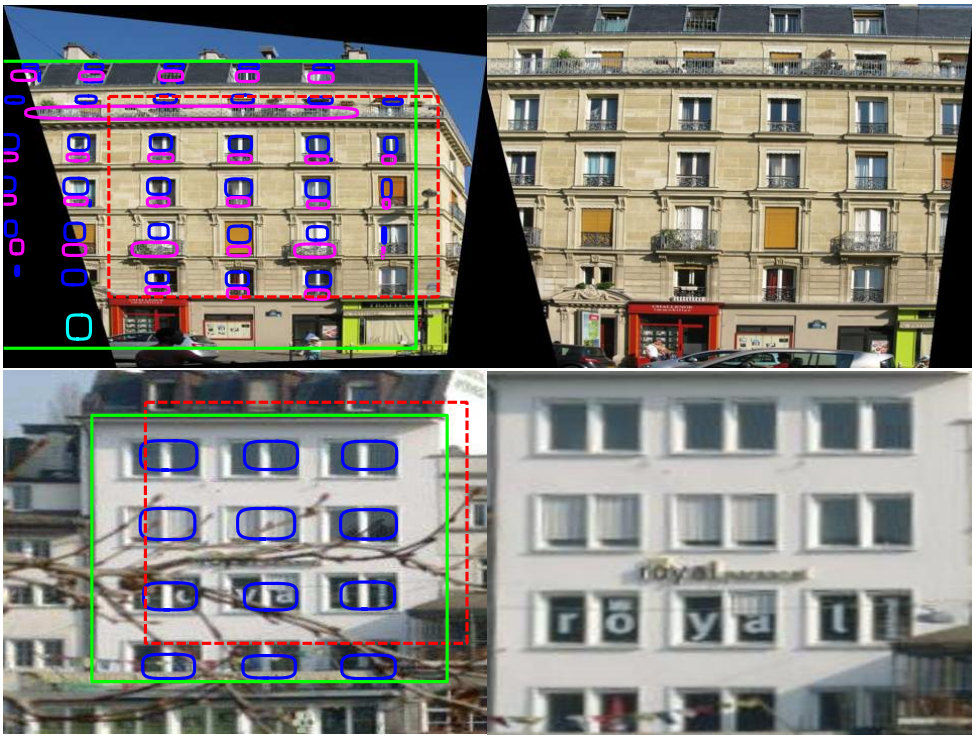


FIGURE 4.11 – Façade recalée initialement (rectangle rouge hachuré) et après convergence (rectangle vert plein) dans une image contenant des parties hors du champ de vision de la caméra (en-haut) ou partiellement occultées (en-bas).

La robustesse de notre méthode aux grands changements d'illumination est particulièrement notable sur la base NancyLights. Le fait d'utiliser la sémantique permet de s'appuyer essentiellement sur la structure géométrique de la façade, les changements d'apparence étant encodés par le réseau de neurones. En particulier, l'invariance du réseau aux changements d'illumination permet de prendre en compte des changements d'éclairage extrêmes, faisant échouer les autres méthodes (voir par exemple la figure 4.9(ligne du bas)).

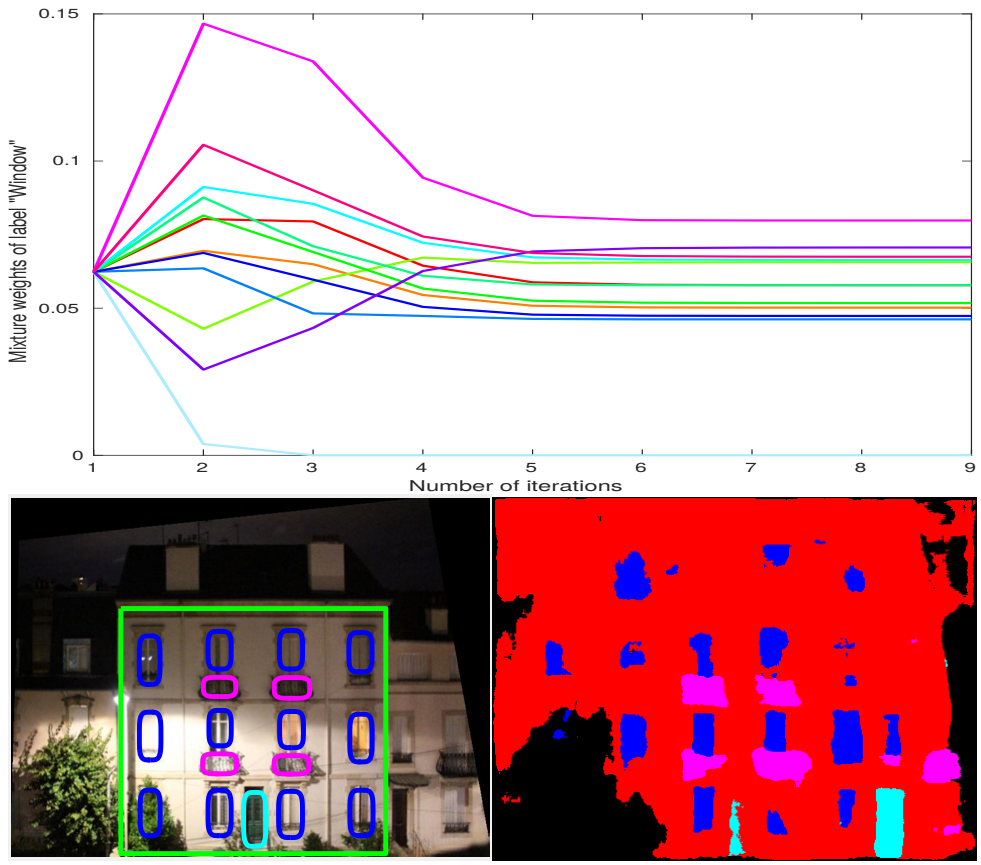


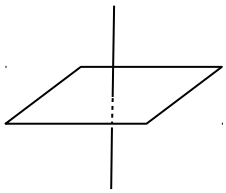
FIGURE 4.12 – Évolution des poids du mélange pour le label “fenêtre” au cours des itérations de l'algorithme EM (en-haut) jusqu'à convergence dans l'image cible (en bas à gauche). Le poids de la fenêtre située en bas à gauche sur la façade, qui est occultée, tend vers zéro (courbe en bleu le plus clair). L'utilisation de la distribution de Dirichlet permet de conserver les autres poids proches de leur valeur initiale malgré la mauvaise qualité de la segmentation *a priori* (en bas à droite).

Pour finir, la figure 4.4 illustre le fait que, bien que la segmentation sémantique *a priori* $P(l_j|i, I)$ ne soit pas mise à jour au cours des étapes EM, les labels l_j peuvent changer d'une itération à l'autre (figure 4.4), grâce à l'influence des gaussiennes généralisées. Des résultats qualitatifs supplémentaires sont présentés dans la thèse d'Antoine Fond [Fon18].

4.6 Conclusion et perspectives

Dans ce chapitre, nous avons montré comment l'utilisation d'un modèle bayésien permet de résoudre le problème du recalage selon une approche corrective tenant compte d'une segmentation sémantique de l'image. L'extension du concept de suivi par synthèse à un modèle de mixture de gaussiennes généralisées permet de bénéficier d'une connaissance approximative de la pose, notamment en présence de structures répétées très fréquentes sur les façades, tout en autorisant une initialisation relativement éloignée de la solution grâce au support infini de ce modèle. Le choix de modéliser les poids de la mixture par une distribution de Dirichlet permet à la méthode EM d'être robuste à la présence d'occultations. L'invariance aux conditions d'illumination du réseau de neurones convolutifs utilisé pour extraire la segmentation sémantique permet enfin d'être robuste à des variations d'éclairage. Dans les faits, cette méthode montre des performances supérieures aux approches classiques sur trois jeux de données illustrant différentes difficultés couramment rencontrées en pratique. Pour ces différentes raisons, l'approche proposée me semble particulièrement intéressante et prometteuse, et je propose en conclusion générale de l'étendre à des scène plus générales.

Une limitation de la méthode non relatée dans les résultats expérimentaux est qu'elle peut échouer dans des cas où la répartition des points échantillonnés dans l'image de référence est quasiment conforme à une distribution uniforme échantillonnée de manière dense (par exemple, une façade densément recouverte de fenêtres). Dans ce cas, la méthode tend à labelliser tous les points comme *outliers*, ou à les attacher à une unique gaussienne lorsque le recalage initial est éloigné de la solution (un exemple est montré en figure 4.10(droite)). La prise en compte conjointe de plusieurs façades, coplanaire ou non, devrait permettre qu'une telle configuration ne se produise que très rarement.



On ne m'estime pas
On ne me rêve pas.

On pose tout sur moi,
On y dessine des figures
Que l'on regarde, moi jamais.

Si pourtant je m'ouvrais ?

Chapitre 5

Odométrie visuelle et modélisation *in situ*

FIGURE 5.1 – Illustrations extraites de l'article *Markerless Tracking using Planar Structures in the Scene*, Gilles Simon, Andrew W. Fitzgibbon et Andrew Zisserman, International Symposium on Augmented Reality, Oct. 2000.

5.1 Introduction

La figure 5.1 est extraite de l'article intitulé "Markerless Tracking using Planar Structures in the Scene", publié à ISAR'2000 [32] et écrit durant mon post-doc à l'Université d'Oxford, dans le groupe *Visual Geometry* dirigé par Andrew Zisserman. Cet article a obtenu un *Lasting Impact Award* à ISMAR'2013. Le jury a en effet considéré qu'il avait eu un impact durable (de plus de dix ans) sur le domaine de la réalité augmentée. Le principe de la méthode décrite dans cet article est le suivant : une région plane, détourée à la main dans la première image d'une séquence (figure 5.1, en haut à gauche), est suivie en temps réel dans les images suivantes, à l'aide de coins de Harris [HS88] appariés par corrélation croisée entre images consécutives de la séquence (le fait que ces images soient proches temporellement permet de chercher le correspondant d'un point de l'image 1 dans son voisinage dans l'image 2, et vice versa). Des homographies inter-images sont calculées de manière robuste par itérations successives entre une étape de RANSAC [FB81] et une étape de raffinement utilisant les appariements retenus lors de la première étape (*inliers* du RANSAC, en vert dans la deuxième image de la figure 5.1). Les quatre sommets d'un rectangle sont également désignés manuellement dans la première image afin de calculer la distance focale de la caméra, ainsi que sa position et son orientation à l'instant initial (figure 5.1, en haut à droite). L'article montre en particulier comment mettre à jour la pose à partir des homographies et de la matrice intrinsèque de la caméra, en vue d'afficher un objet virtuel, fermement ancré à la scène dans toutes les images de la séquence (ligne du bas en figure 5.1).
Cette méthode semble aujourd'hui très simple. Elle marque néanmoins le début de la RA temps réel basée sur le suivi automatique d'indices naturellement présents dans la scène. Avant cette publication, les seuls systèmes de RA temps réel reposaient sur la présence, soit de marqueurs artificiels placés dans le champ de vision de la caméra [KB99], soit de capteurs physiques de mouvements (magnétiques, inertiels, ...) solidaires de la caméra [ALJ+99]. Des algorithmes de SFM [FZ98] étaient déjà couramment utilisés, notamment dans des logiciels destinés à la postproduction cinématographique, tels que Maya Live©. Ceux-ci reposaient toutefois sur une géométrie multioculaire, inexploitable tant que plusieurs vues relativement distantes de la scène ne sont pas disponibles, et une étape d'ajustement de faisceau très coûteuse en temps de calcul. Des adaptations de notre algorithme, utilisant notamment des primitives de type SIFT au lieu de Harris, ont été implémentées dans plusieurs *toolkits* de RA tels que celui de Metaio© ou encore ARToolkit© depuis la version 5.
Dans la lignée de cet article, les travaux présentés dans ce chapitre exploitent la présence dans l'environnement de plans texturés. Dans la thèse de Javier-Flavio Vigueras Gomez [VG07], nous avons mesuré l'impact d'erreurs de calibration de la caméra sur le calcul de pose en contexte de RA [26, 25]. Ce travail a obtenu un *best paper* à ISMAR'2005 mais j'ai choisi de ne pas le développer plus en avant dans ce mémoire. Une autre contribution de la thèse de Flavio a été de permettre un calcul de pose plus stable (sans effet de tremblement de la scène virtuelle par rapport à l'image réelle), en opérant une sélection de modèle de mouvement [28] (section 5.2). La capacité de faire la distinction entre un mouvement stationnaire de caméra, une rotation pure et un déplacement rigide à six degrés de liberté m'a permis de réaliser un système de modélisation *in situ* [20], autorisant la confrontation en temps réel du modèle 3D en cours d'élaboration avec la réalité terrain *via* un affichage en réalité augmentée (section 5.3).

5.2 Sélection du modèle de mouvement

Un phénomène couramment observé par les utilisateurs de systèmes de RA, est que les objets virtuels semblent parfois "trembler" par rapport à la scène réelle, en particulier lorsque le mouvement de la caméra est lent. Ce phénomène apparaît aussi bien dans le cas où la pose est calculée à partir de données capteurs (voir par exemple la vidéos 3 [1]) que dans le cas où elle est calculée en utilisant la vision par ordinateur (voir par exemple la vidéo 6). La raison du phénomène est

1. Nous rappelons que les vidéos associées au mémoire sont disponibles à l'adresse `https://members.loria.fr/GSimon/habilitation-a-diriger-des-recherches/`.

identique dans les deux cas : le bruit sur les données en entrée de l'algorithme utilisé se traduit par des petites erreurs, discontinues, sur les paramètres de la pose calculés dans chaque image.

5.2.1 État de l'art et contributions

Par le passé, plusieurs travaux ont préconisé d'utiliser un filtre de Kalman pour prédire et stabiliser la trajectoire de la caméra [ALJ+99, RD06a]. Un tel filtre n'est cependant pas toujours avantageux pour la RA, en particulier parce que les modèles dynamiques considérés en général (modèle à vitesse ou accélération constante) ne sont pas satisfaisants pour décrire les mouvements humains. Cette solution est à peu près la seule envisageable pour stabiliser le calcul de pose lorsque des capteurs physiques de mouvement sont utilisés. Dans ce cas la pose est obtenue en quelque sorte "a l'aveugle", car indépendamment de l'image dans laquelle les augmentations auront lieu. L'avantage de la vision par ordinateur est qu'elle s'appuie sur l'image elle-même pour calculer la pose par recalage d'un modèle 3D (dans notre cas, un ou plusieurs plans). Il est donc possible de confronter les mesures de l'image avec les mesures prédites par la solution obtenue (erreur de reprojection ou de transfert dans le cas d'une homographie), et de tirer profit de cette connaissance pour stabiliser la trajectoire de la caméra.
Dans la lignée des travaux de Matsunaga et Kanatani [MK00] et de Torr et al. [TFZ98], nous avons ainsi cherché à utiliser des méthodes de sélection de modèle pour réduire les fluctuations sur les paramètres calculés et améliorer ainsi l'impression visuelle de la scène augmentée. L'idée sous-jacente à la sélection de modèles est qu'un modèle d'ordre élevé approche toujours mieux un ensemble de données qu'un modèle d'ordre inférieur. Cependant, les modèles d'ordre élevé approximent aussi une partie du bruit qu'ils sont censés éliminer. Un modèle d'ordre élevé, bien que théoriquement plus précis, est donc en fait moins stable aux perturbations aléatoires des données. Une bonne méthode de sélection de modèles doit donc réaliser un compromis entre précision et stabilité. Le principe des méthodes de sélection de modèles est d'exiger que le modèle choisi explique bien les données tout en étant le plus simple possible.
De nombreux critères de sélection de modèles ont été proposés dans la littérature [BS98]. La plupart d'entre eux sont basés sur des critères statistiques ou sur des critères issus de la théorie de l'information. Parmi eux, les plus utilisés sont sans doute le critère d'Akaike (AIC) ainsi que le critère de description de longueur minimale (MDL). Le critère AIC peut être vu comme une approximation d'un critère entropique (la distance de Kullbak-Leibler), alors que le critère MDL favorise le modèle dont la description en terme de bits est minimale. Quel que soit le critère considéré, la valeur associée à chaque modèle est calculée comme la somme du résidu $\hat{J}$ et d'une fonction du facteur de complexité du modèle M_k, destiné à pénaliser les modèles plus complexes :

$$E_{critere} = \hat{J} + \hat{\epsilon}^2 c(M_k),$$

où $\hat{\epsilon}$ représente le niveau de bruit, qui peut être estimé en fonction du résidu [2].
La sélection de modèle a été utilisée par Kanatani [MK00] pour résoudre le problème de calibration d'une caméra. Dans leur approche, seuls deux critères de sélection sont étudiés, AIC et $gMDL$, en considérant un seul motif plan pour le recalage.
Dans nos travaux, nous avons repris cette idée et apporté les contributions suivantes :

1. plusieurs plans texturés, ne présentant pas de motif particulier, sont considérés
2. une étude de performance est proposée, comprenant plusieurs critères de sélection,
3. la persistance temporelle du choix d'un modèle est prise en compte.

5.2.2 Suivi multiplan

La prise en compte de plusieurs plans de la scène, au lieu d'un seul, est évidemment intéressante, à la fois pour améliorer la cohérence spatiale des incrustations 3D (qu'elles ne soient pas seulement pertinente localement), et pour palier les éventuels problèmes d'occultation, d'angle de vue réduit

2. Par exemple, $\epsilon^2 = \frac{\hat{J}}{2-7/n}$, où n est le nombre de données, est utilisé dans [MK00].

Critère	Terme de complexité
Akaike AIC [Aka74]	$2k$
Bozdogan CAIC [Boz87]	$k(\log n + 1)$
Bozdogan CAICF [Boz87]	$k(\log n + 2) + \log \lvert\mathbf{I}(\theta_k)\rvert$
Schwarz BIC [Sch78]	$2k \log n$
Kanatani gMDL [MK00]	$-k \log \hat{\epsilon}^2$

TABLE 5.1 – Termes de complexité $c(M_k)$ des critères de sélection évalués. k est le nombre de paramètres du modèle M_k (représentés par le vecteur θ_k) et n le nombre de données utilisées pour estimer ces paramètres.

ou de sortie de champ temporaires des régions planes. La méthode proposée pour exploiter le suivi de plusieurs plans dont les équations sont connues est décrite dans [31] et illustrée par les vidéos 5 et 6. Plusieurs méthodes de résolution ont été comparées, dont la plus précise consiste à utiliser la méthode de Levenberg-Marquardt pour minimiser la fonction de coût suivante :

$$J(\theta) = \frac{1}{N_1 + \cdots + N_n} \sum_{k=1}^{n} \sum_{j=1}^{N_k} |\mathbf{x'}_{kj} - Z(\mathbf{H}_k \mathbf{x}_{kj})|^2, \tag{5.1}$$

où θ est le vecteur des paramètres de la pose (translation en x, y, z et angles d'Euler dans le cas général), n est le nombre de plans, N_k le nombre de paires de points $(\mathbf{x}_{kj}, \mathbf{x'}_{kj})_{1 \leq j \leq N_k}$ en correspondance pour le plan k, $\mathbf{H}_k$ l'homographie induite par le plan k [3], et Z la fonction qui divise les coefficients d'un vecteur de taille 3 par le dernier coefficient (passage des coordonnées homogènes aux coordonnées cartésiennes). Les paramètres initiaux de cette optimisation sont donnés par la pose calculée dans l'image précédente. La valeur de la fonction $J(\theta)$ à la fin de l'optimisation est le résidu $\hat{J}$ utilisé dans ce chapitre.

5.2.3 Critères de sélection

La sélection de modèle a été exploitée pour résoudre divers types de problèmes, tels que la fusion de données surfaciques [BS98], la sélection de mouvement en géométrie bi ou multioculaire [Tor97, TFZ98] ou encore la détection de primitives géométriques [Kan02]. Il s'avère qu'aucun critère de sélection ne peut être reconnu comme meilleur dans tous les cas. Dans notre étude, nous en avons comparé plusieurs, résumés en table 5.1. Le critère CAICF utilise la matrice d'information de Fisher $\mathbf{I}(\theta) = E\left(\frac{\partial}{\partial\theta} J(\theta)^t \frac{\partial}{\partial\theta} J(\theta)\right)$.
Notre évaluation repose sur des tests synthétiques générés à partir d'une séquence de mire à trois plans (figure 5.2). Plus précisément, une table micrométrique est utilisée pour faire subir à la caméra les mouvements décrits en table 5.2. Les poses sont alors calculées en utilisant la méthode de Toscani [Tos87], à l'aide des centres des disques blancs présents sur la mire, facilement détectables dans les images, et dont les coordonnées 3D sont données par le fabriquant. Les données synthétiques sont obtenues en reprojetant les points 3D à partir des poses calculées, puis en ajoutant du bruit gaussien sur les coordonnées 2D obtenues. Pour chaque image i, nous calculons la pose en minimisant le résidu exprimé dans l'équation (5.1) par rapport aux paramètres du modèle de mouvement considéré : stationnaire, rotation pure ou mouvement rigide à six degrés de liberté. Afin d'éviter que des problèmes éventuels de dérive ne perturbent l'évaluation des modèles, nous utilisons la vérité terrain du point de vue dans l'image $i-1$ comme estimée initiale de l'optimisation dans l'image i. Le véritable modèle étant connu, nous montrons en tables 5.3 et 5.4 le pourcentage de modèles correctement choisis pour chacun des critères évalués, pour un bruit d'écart-type $\sigma = 0.3$ et un bruit d'écart-type $\sigma = 1$, respectivement. La colonne "+ complexe" indique la proportion de cas où un modèle d'ordre plus élevé que le vrai modèle a été choisi, tandis que la colonne "- complexe" indique la proportion de cas où un modèle d'ordre moins élevé, donc trop restrictif, a été choisi.

3. $\mathbf{H}_k = \mathbf{K}(\mathbf{R} - \mathbf{t}\mathbf{v}_k^T/d_k)\mathbf{K}^{-1}$, avec $\mathbf{K}$ la matrice intrinsèque de la caméra, $\mathbf{R}, \mathbf{t}$ la matrice de rotation et, resp., le vecteur de translation entre les deux vues, $\mathbf{v}_k$ la normale au plan k et d_k sa distance à l'origine.

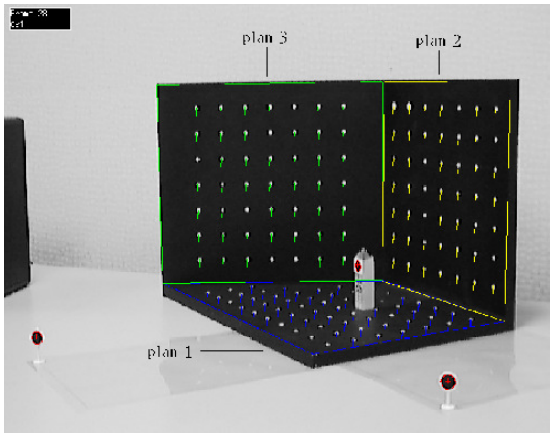


FIGURE 5.2 – Mire de calibration utilisée pour comparer les critères de sélection.

# Image	Mouvement
0 - 19	Rotation et Translation
20 - 29	Stationnaire
30 - 39	Translation selon l'axe des Y
40 - 49	Translation selon l'axe des X
50 - 64	Panoramique
65 - 69	Stationnaire

TABLE 5.2 – Description des mouvements de la caméra au cours de la séquence de la mire.

Pour une valeur de bruit modérée (table 5.3), la plupart des critères se comportent relativement bien, et un modèle d'ordre supérieur ou égal au véritable modèle est presque toujours choisi. On note cependant que les critères AIC et $gMDL$ ont tendance à produire des modèles trop généraux, ce qui n'est pas très intéressant dans une optique de stabilisation.
Lorsque le bruit augmente (table 5.4), les performances de certains critères se dégradent nettement. Cependant on peut noter que les deux critères CAIC et CAICF se comportent le mieux : ils se comportent très bien dans le cas stationnaire, n'ont que peu tendance à sélectionner un modèle trop général dans le cas panoramique et se comportent très honorablement dans le cas général. Le critère CAICF induit par ailleurs moins de sélections de modèles d'ordre inférieur que CAIC. Ces résultats tendent donc à montrer que l'introduction de la matrice d'information sur les paramètres calculés améliore la sélection de modèle. Pour le lecteur intéressé, des résultats plus complets sont présentés dans la thèse de Flavio [VG07]. Dans la suite du chapitre, nous utilisons le critère CAICF.

5.2.4 Cohérence temporelle

La sélection de modèle améliore très sensiblement la stabilité des points de vue et la qualité visuelle des incrustations. Cependant, il peut subsister des erreurs dans l'étiquetage du mouvement, par exemple dues à une confusion entre un mouvement panoramique et une translation dans le cas de petits mouvements. La répétition de choix inappropriés de mouvements pouvant conduire à faire diverger la pose, nous souhaitons améliorer le processus de sélection. Pour cela, nous proposons d'utiliser la cohérence temporelle des modèles détectés en utilisant plus de deux vues pour valider la sélection du modèle, ce qui présuppose qu'un modèle de mouvement persiste dans au moins trois images consécutives. Dans le cas où des modèles différents seraient trouvés, le modèle le plus général d'entre eux sera choisi, sachant qu'il est préférable de choisir un modèle plus général plutôt qu'un modèle plus restrictif qui peut conduire à la divergence. Plus précisément, notre algorithme est le suivant :

1. Les paramètres de la caméra sont connus pour les images $i-1$ et $i-2$.

mouvement	critère	$\sigma = 0.3$		
		- complexe	**correct**	+ complexe
statique	AIC	-	83.1%	16.9%
	CAIC	-	98.7%	1.3%
	CAICF	-	100.0%	0.0%
	BIC	-	100.0%	0.0%
	gMDL	-	77.5%	22.5%
pano	AIC	0.0%	85.3%	14.7%
	CAIC	0.0%	99.3%	0.7%
	CAICF	0.0%	98.7%	1.3%
	BIC	0.0%	100.0%	0.0%
	gMDL	0.0%	84.7%	15.3%
général	AIC	0.0%	100.0%	-
	CAIC	1.5%	98.5%	-
	CAICF	1.3%	98.7%	-
	BIC	5.4%	94.6%	-
	gMDL	0.0%	100.0%	-

TABLE 5.3 – Résultats de la sélection de modèle pour le niveau de bruit $\sigma = 0.3$.

mouvement	critère	$\sigma = 1.0$		
		- complexe	**correct**	+ complexe
statique	AIC	-	83.7%	16.3%
	CAIC	-	100.0%	0.0%
	CAICF	-	100.0%	0.0%
	BIC	-	100.0%	0.0%
	gMDL	-	0.0%	100.0%
pano	AIC	0.0%	86.7%	13.3%
	CAIC	0.0%	100.0%	0.0%
	CAICF	0.0%	97.3%	2.7%
	BIC	0.0%	100.0%	0.0%
	gMDL	0.0%	0.0%	100.0%
général	AIC	11.5%	88.5%	-
	CAIC	24.1%	75.9%	-
	CAICF	20.3%	79.7%	-
	BIC	33.6%	66.4%	-
	gMDL	0.0%	100.0%	-

TABLE 5.4 – Résultats de la sélection de modèle pour le niveau de bruit $\sigma = 1$.

2. Sélectionner le modèle de mouvement $M_{i,i-1}$ entre l'image courante i et la précédente $i-1$.

3. Sélectionner le modèle de mouvement $M_{i,i-2}$ entre l'image courante i et l'image $i-2$.

4. Le modèle sélectionné est le modèle M' le plus simple tel que l'espace des paramètres de $M_{i,i-1}$ et $M_{i,i-2}$ soient des sous espaces de M'. Si les espaces sont emboîtés, ceci signifie que M' sera le plus général des deux modèles $M_{i,i-1}$ et $M_{i,i-2}$.

Cette méthode comporte toutefois un léger inconvénient quand on passe d'un modèle complexe à un plus simple. La première image de la transition est alors affectée au modèle le plus complexe, et la transition au modèle le plus simple se fait avec un temps de retard.

Enfin, une fois le modèle estimé sur la base de trois images, la position de la caméra courante est recalculée de la façon suivante en tenant compte des points en correspondance $(\mathbf{x}_{i-1}, \mathbf{x}_i)$ et

$(\mathbf{x}_{i-2}, \mathbf{x}_i)$:

$$\begin{aligned} J(\theta) &= \sum_{k=1}^{n}\sum_{j=1}^{N_k} |\mathbf{x}_{kj}^{i} - Z(\mathbf{H}_k^{i,i-1}\mathbf{x}_{kj}^{i-1})|^2 \\ &+ \sum_{k=1}^{n}\sum_{j=1}^{N_k} |\mathbf{x}_{kj}^{i} - Z(\mathbf{H}_k^{i,i-2}\mathbf{x}_{kj}^{i-2})|^2. \end{aligned}$$

5.2.5 Quelques résultats

Nous présentons dans cette section quelques résultats de calcul de pose utilisant la méthode proposée ci-dessus. Nous montrons les résultats obtenus sur la séquence synthétique déjà utilisée, ainsi que sur une séquence réelle.

5.2.5.1 Séquence synthétique

Nous présentons tout d'abord les résultats obtenus sur la séquence synthétique bruitée ($\sigma = 1.0$). La figure 5.3 montre les modèles sélectionnés selon que deux ou trois images sont utilisées pour décider. 0 indique un modèle stationnaire, 1 une rotation pure, 2 un mouvement général. Nous pouvons constater qu'entre les images 30 et 50, si nous utilisons seulement deux images pour la sélection, il y a un certain nombre de confusions entre modèle panoramique et modèle général. Ces problèmes s'atténuent visiblement quand les triplets d'images sont utilisés pour la sélection. De manière générale, la probabilité de sélectionner un mauvais modèle décroît quand les triplets sont utilisés. La figure 5.4 montre la composante translationnelle t_z calculée. Elle indique que l'utilisation de triplets améliore aussi la précision du point de vue calculé par rapport à l'utilisation de deux vues.

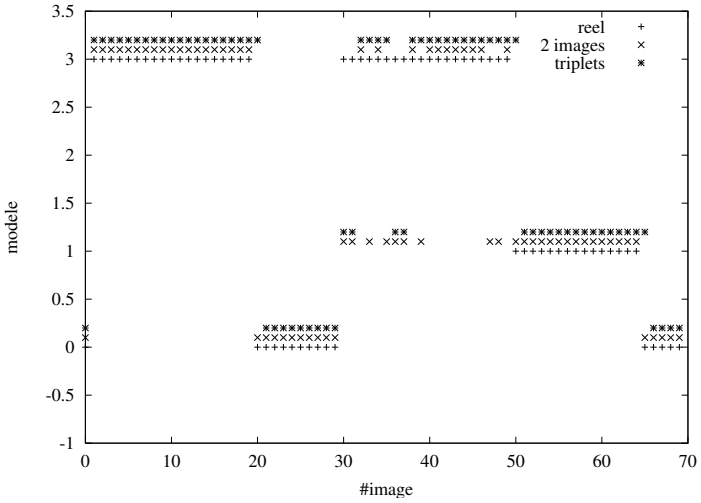


FIGURE 5.3 – Sélection du modèle en utilisant deux ou trois vues sur la séquence synthétique ($\sigma = 1.0$).

5.2.5.2 Séquences réelles

Nous considérons à présent une séquence de 225 images d'une maquette en carton représentant une pièce d'intérieur (ces travaux ont été réalisés dans le cadre du projet européen ARIS). Cette séquence est constituée des mouvements successifs suivants : stationnaire, général, stationnaire, panoramique, stationnaire, panoramique et enfin stationnaire.
La figure 5.5 montre que l'utilisation de triplets améliore légèrement la sélection du modèle. Les images de 30-110 montrent par exemple que certains mouvements, identifiés à tort comme

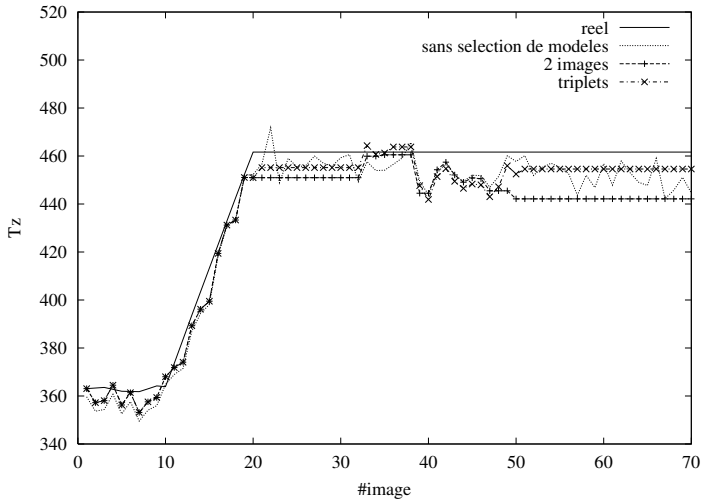


FIGURE 5.4 – Composante t_z calculée en utilisant la sélection sur deux ou trois images pour la séquence synthétique.

panoramiques ou stationnaires lorsque deux vues sont utilisées, sont correctement classifiés en modèle général avec trois vues.
La figure 5.6 montre la composante t_x de la translation obtenue entre les images 0 et 50 de la séquence (mouvement stationnaire suivi d'un mouvement général), selon que le modèle général est toujours utilisé, ou la sélection de modèle à 2 ou 3 images. Ce graphique révèle l'effet stabilisateur de la sélection de modèle puisque la section 0-30 est bien stationnaire dans le cas où la sélection de modèle est utilisée. Les images de la figure 5.7 montrent également l'impact de la sélection de modèle sur une intégration 3D : l'image (a) montre la scène augmentée au bout de 200 images lorsqu'aucune sélection de modèle n'est utilisée (c'est à dire quand le modèle général est toujours utilisé), l'image (b) montre l'image augmentée quand la sélection est utilisée. Il est clair sur ces deux images que la sélection apporte de la stabilité et donc de la précision, le cube étant sensé rester aligné avec les motifs rectangulaires dessinés au sol. Le lecteur pourra s'en convaincre en comparant la vidéo 8, obtenue en utilisant la sélection de modèle, à la vidéo 7, utilisant le modèle général uniquement. De façon générale, la sélection de modèle réduit les variations aléatoires de certains paramètres du point de vue, et ainsi l'accumulation d'erreur obtenue par composition des mouvements relatifs.

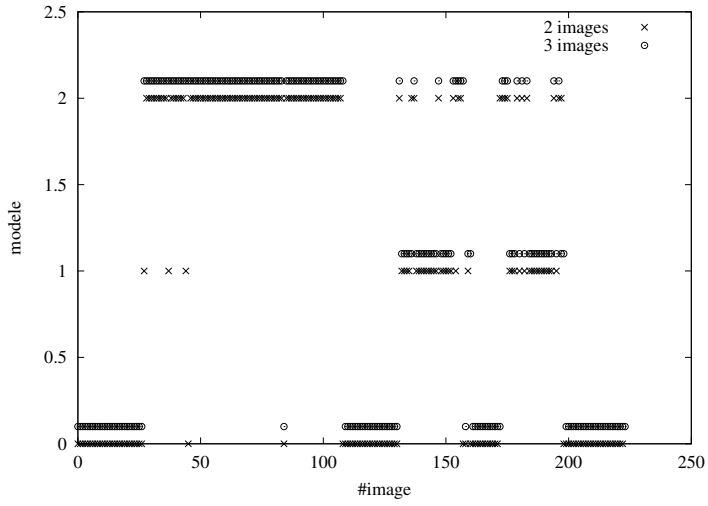


FIGURE 5.5 – Séquence réelle : sélection du modèle.

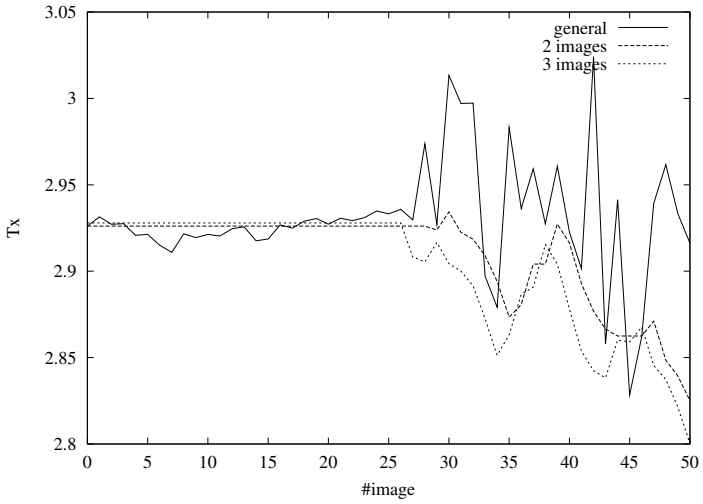


FIGURE 5.6 – Séquence réelle : translation en x calculée avec ou sans sélection de modèle.

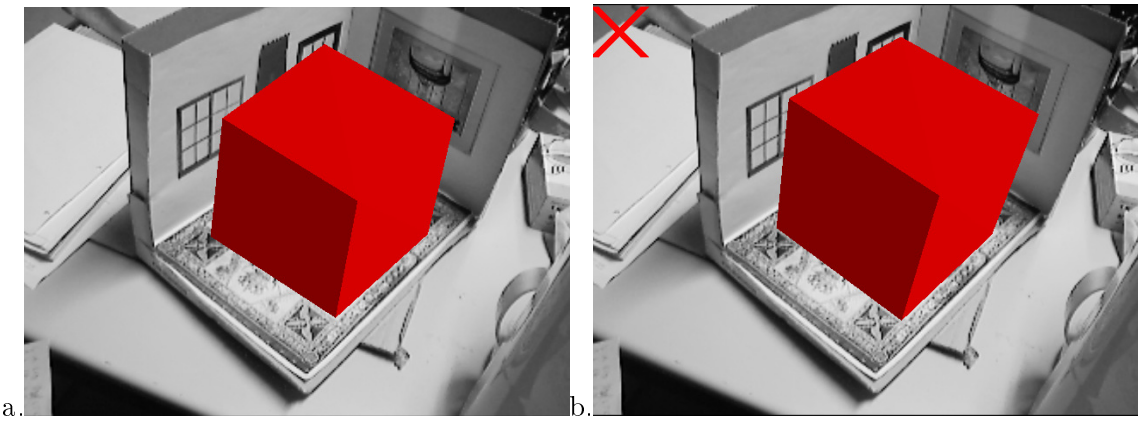


FIGURE 5.7 – Un cube incrusté sur l'image 200 : (a) sans sélection de modèle ; (b) : avec sélection

La vidéos 9 montre enfin la sélection de modèle à l'œuvre dans une séquence d'images acquises dans un hall du Loria. La vidéo 10 montre une vue de dessus des mouvements de caméra calculés, et la vidéo 11 une intégration 3D d'un billard virtuel dans la séquence d'images. On constate que celui-ci semble bien ancré au sol.

5.3 Application à la modélisation *in situ*

Si notre méthode de sélection de modèle permet d'éviter les phénomènes de tremblement, et dans une certaine mesure, de dérive, propres à de nombreux systèmes de RA, elle présente un intérêt direct dans certaines applications de vision par ordinateur, telles que la reconstruction multioculaire, dont les étapes de triangulation ou d'ajustement de faisceau requièrent des parallaxes non nulles. Que cette reconstruction soit calculée hors ligne à partir de plusieurs image (SFM) ou à la volée (SLAM), elle peut être corrompue par la présence de mouvements singuliers (stationnaires ou rotationnels) au sein de la séquence traitée. La sélection de modèle permet alors de supprimer les cas singuliers de cette séquence, ou de leur appliquer un traitement particulier. Inversement, la reconstruction monoculaire opère dans une image isolée, ou dans plusieurs images prises depuis une position fixe et assemblées en panoramique. Reconstruire une scène de manière interactive en utilisant une caméra en mouvement, comme nous le proposons dans cette section, requiert donc de pouvoir séparer les mouvements généraux des deux autres types de mouvement.

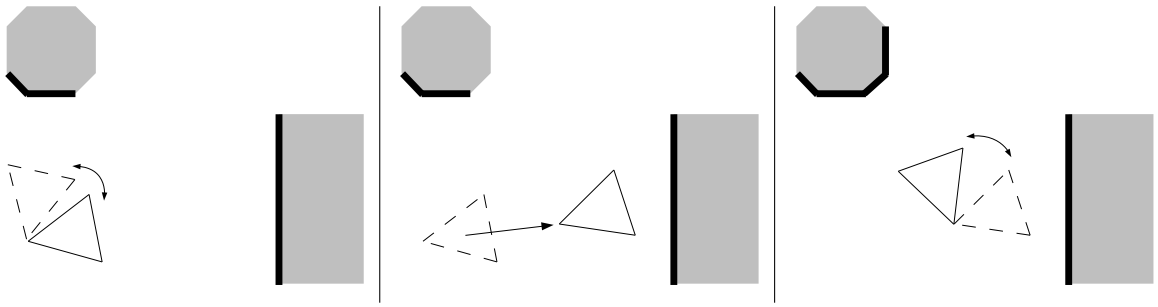


FIGURE 5.8 – Principe de fonctionnement de notre système de modélisation *in situ* (schéma en vue de dessus) : des rotations pures de la caméra (en mode modélisation – les traits épais représentent les faces modélisées) alternent avec des déplacements à six degrés de liberté permettant d'accéder aux parties initialement non visibles de la scène. Les faces modélisées servent à calculer la pose au cours des déplacements, par suivi de plans texturés.

5.3.1 État de l'art et contributions

La méthode proposée dans cette section peut être vue aussi bien comme :

1. une méthode de SLAM visuel, autorisant la modélisation interactive, à la volée, des structures polyédriques utiles à la fois au calcul de pose et au positionnement des éléments virtuels,
2. une méthode de modélisation *in situ*, permettant de reconstruire une scène 3D en temps réel, en se déplaçant à l'intérieur ou autour de la scène de manière à rendre visible les éléments à ajouter au modèle, non visibles en position initiale.

5.3.1.1 SLAM visuel

Le SLAM visuel [DM02, CCC+16] permet de construire automatiquement une carte de l'environnement, le plus souvent sous forme d'un nuage de points, en même temps que cet environnement est exploré. Les points reconstruits servent à calculer les poses suivantes, tandis que les poses calculées servent à reconstruire des points supplémentaires. Notre système repose sur le même principe que ce schéma : l'utilisateur alterne des phases d'arrêt, pendant lesquelles il modélise les éléments de la scène visibles depuis le lieu où il se trouve, en faisant pivoter la caméra, avec des phases de déplacement lui permettant d'atteindre d'autres parties de la scène (figure 5.8). Les faces des objets polyédriques modélisés à l'arrêt sont utilisées pour calculer les mouvements de la caméra lors des déplacements. La sélection de modèle permet au système de discerner les phases d'arrêt (comprenant des mouvements stationnaires et rotationnels) des phases de déplacement, à six degrés de liberté. Dans cette interprétation, on peut par exemple imaginer une application destinée à des architectes paysagistes ou d'intérieur, leur permettant d'esquisser rapidement, sur place, un projet d'aménagement ou de décoration et de montrer aussitôt le résultat au client en réalité augmentée.

5.3.1.2 Modélisation *in situ*

La modélisation 3D offre un vaste champ d'applications notamment dans les domaines de l'architecture, de l'infographie, de la RA, de la RV, de la robotique, de la géomatique, etc. La méthode que nous proposons permet de modéliser une scène directement par-dessus la réalité, ce qui évite de prendre des mesures dans la scène (une seule mesure doit être prise si la taille du modèle doit être connue dans une unité particulière) et permet de confronter en temps réel la géométrie reconstruite à la réalité. Plusieurs systèmes ont déjà été proposés pour modéliser une scène à partir d'une photo [DTM96] (comme dans le logiciel Trimble SketchUp©), de plusieurs photos (comme dans le logiciel Realviz ImageModeler©) ou d'une vidéo [vdHDT+07]. Mais, une fois les images acquises et importées dans le logiciel, rien ne garantit que toutes les parties du modèle soient bien visibles dans les images, ni même que toutes les images soient exploitables. Il n'est pas

rare alors de devoir retourner sur le site pour acquérir de nouvelles vues, ce qui peut engendrer des coûts supplémentaires. Il peut arriver aussi qu'une scène soit transformée après la première acquisition, voire plus accessible du tout. Un décor de cinéma par exemple, est fréquemment mis en place le matin, puis démonté à tout jamais le soir même. Notre système de modélisation *in situ* permet de court-circuiter le délai entre acquisition et exploitation des données. Le modèle en cours de construction est reprojeté en temps réel dans les images acquises, et ce processus se poursuit tant que le modèle n'est pas complet. En supplément, le modèle reconstruit peut être réutilisé à des fins de RA où d'intégration 3D en postproduction, celui-ci ayant déjà fait ses preuves quant à son exploitabilité pour le calcul de pose.
La modélisation *in situ* n'a fait l'objet que d'un petit nombre de travaux dans la littérature. Les premiers auteurs à avoir envisagé une telle approche sont Wayne Piekarski et Bruce H.Thomas de l'Université de South Australia, en 2001 [PT01]. Le terme employé alors était "*construction at a distance*". Le principe était de permettre à un utilisateur de modéliser des objets de la scène à travers une interface de RA, en assemblant des primitives élémentaires par CSG (*constructive solid geometry*). Un gant de données était utilisé pour l'interaction et la pose était obtenue à l'aide de capteurs inertiels et d'un GPS (voir la vidéo 4). Ce système était assez complexe du point de vue des interactions et peu précis du fait de la technologie utilisée pour le suivi de pose. Un système de modélisation interactive purement basé image a été proposé par Bunnun et al. [BMC08]. Ce système a pour point commun avec le notre que la caméra elle-même sert de dispositif d'interaction (nul besoin de souris, gant de données ou autre périphérique). Il présente cependant plusieurs aspects contraignants : 1. un marqueur est utilisé pour démarrer le processus, 2. la méthode utilise des techniques de stéréovision, qui nécessitent de déplacer sans arrêt la caméra, ce qui peut être aisé lorsqu'on modélise de petits environnements (des objets posés sur un bureau) mais difficile à l'échelle d'un bâtiment par exemple, 3. les interactions sont assez complexes à mettre en œuvre (par exemple, une action demandée est de ramener le curseur-caméra à l'extrémité d'un segment en suivant une droite épipolaire).
À la différence de cette technique, notre méthode permet de modéliser des objets relativement complexes à n'importe quelle échelle, en utilisant des briques de construction très simples. En fait, nous reprenons les briques de cosntruction du logiciel Sketchup, devenu célèbre [4] en grande partie grâce à sa facilité de prise en main.

5.3.2 Description générale de la méthode

La méthode proposée repose sur une alternance entre deux phases opératoires (figure 5.8) :

Une phase de modélisation. La scène (ou une partie de la scène) est modélisée de manière interactive, selon le principe de la "caméra-souris" qui remplace les mouvements de curseurs dans une image fixe par des mouvements d'image (rotations pures) avec un curseur fixe (voir la section 5.3.3). L'homographie inter-images est calculée selon la méthode décrite dans [32] utilisant des coins de Harris détectés dans toute l'image. Une procédure de reprise automatique est utilisée pour calculer la pose en cas d'échec de suivi, détecté par seuillage sur le nombre d'inliers du RANSAC. Cette procédure est basée sur des appariements de descripteurs SIFT. Plus précisément, lorsque l'utilisateur entre en phase de modélisation (c'est-à-dire au début du processus ou lorsqu'il cesse de se déplacer) des points de SIFT sont détectés dans toute l'image. Ces points sont transférés d'image en images en utilisant les homographies calculées à l'aide des coins de Harris (une détection des points de SIFT dans toutes les images alourdirait considérablement les calculs). L'image est divisée en une grille de taille 3×3, et lorsqu'un rectangle de la grille est vide après que les points de SIFT aient été transférés, une nouvelle détection de points de SIFT est effectuée à l'intérieur du rectangle (voir le début de la vidéo 13). Il est à noter que ce mode peut être utilisé seul, contribuant déjà de manière intéressante à l'état de l'art de la modélisation monoculaire, dans la mesure où, premièrement, la possibilité de

4. Chaque seconde, Sketchup est lancé 30 fois à travers le monde (statistique communiquée en 2019 par la société Trimble qui commercialise ce logiciel).

réorienter la caméra offre un champ de vision bien plus grand (jusqu'à 360 deg) que celui de la caméra elle-même et, deuxièmement, les manipulations à la souris sont remplacées par des mouvements de caméra, ce qui s'avère beaucoup plus pratique avec des dispositifs mobiles tels qu'un téléphone portable ou des lunettes de RA.

Une phase de déplacement. Lorsqu'au moins une face de la scène a été modélisée et que l'utilisateur se déplace, les mouvements de caméra sont calculés par suivi de plan à 6 degrés de liberté selon la méthode décrite en section 5.2.2, utilisant des coins de Harris. Cette phase permet à l'utilisateur de se rapprocher de certaines parties de la scène, ou de rendre visibles de nouvelles faces pour poursuivre la modélisation. Dans ce mode, la géométrie modélisée précédemment doit rester partiellement visible afin de permettre le suivi de mouvement, ce qui n'est pas le cas en mode modélisation. Une procédure de reprise automatique peut être utilisée afin de reprendre le calcul de pose en cas d'échec de suivi (détecté à partir du nombre d'inliers du RANSAC) ou à la demande de l'utilisateur. Des descripteurs SIFT sont utilisés à cette fin, calculés et mémorisés à chaque création de face (texturée) et calculés dans l'image courante quand la reprise est demandée.

Le système démarre en mode modélisation. Une fois que la géométrie 3D a été initialisée, l'alternance entre modélisation et suivi de déplacements se fait automatique en utilisant la sélection de modèle.

5.3.3 Interactions pour la modélisation

Notre système peut être vu comme une version immersive du logiciel SketchUp, dont les principes d'interaction sont décrits par exemple dans [OSD05]. Ce logiciel est à mi-chemin entre l'esquisse au crayon et la CAO (Conception Assistée par Ordinateur). Une fonctionnalité proposée assez tôt dans ce logiciel est la possibilité d'aligner les axes du modèle avec des directions orthogonales de la scène (désignation manuelle d'un repère de Manhattan tel que décrit au chapitre 2), afin de modéliser la scène par-dessus une photographie. Pour ce faire, des points désignés à la souris sur la photographie sont reconstruits en 3D par intersection du rayon inverse avec la géométrie définie précédemment (le plan au sol par défaut). Ces fonctionnalités sont reprises dans notre système, avec la principale différence que nous considérons des images dynamiques au lieu d'images statistiques. De plus, la caméra est utilisée comme dispositif d'interaction au lieu d'une souris, ce qui, comme nous l'avons déjà souligné, est particulièrement adapté aux dispositifs mobiles.

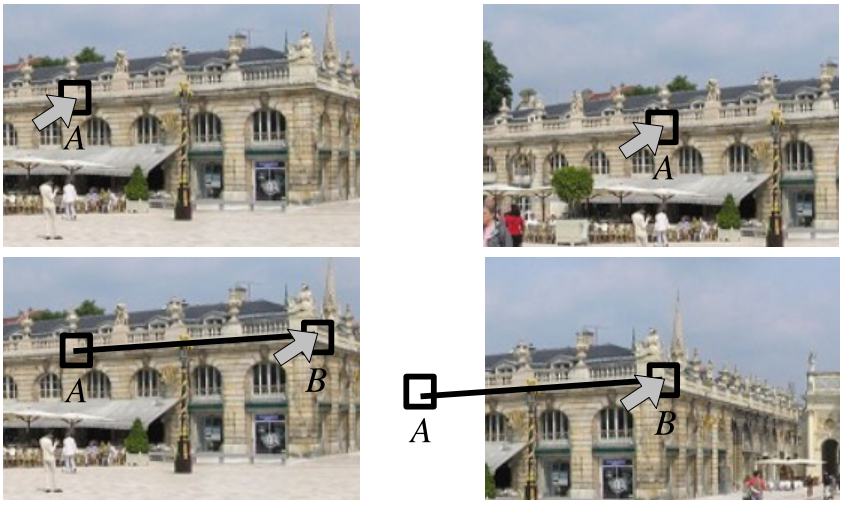


FIGURE 5.9 – Équivalence entre curseur mobile / caméra fixe (à gauche) et curseur fixe / caméra mobile (à droite).

Une idée clé de notre méthode est que les interactions peuvent être effectuées avec la caméra. En effet, considérons que l'utilisateur veuille tracer un segment entre deux points physiques A et B

de la scène acquise. Il existe deux manières équivalentes de réaliser cela (figure 5.9) : 1. la caméra reste fixe et la souris est utilisée pour déplacer un curseur du point A au point B (colonne de gauche) ou 2. la caméra est manipulée de manière à ce que le point A puis le point B se retrouve successivement sous un point fixe de l'image, par exemple son centre (colonne de droite). La seconde solution, utilisée dans notre implémentation, ne requiert pas d'utiliser une souris. En revanche, la position du point A doit être mise à jour à chaque instant pendant que l'utilisateur réoriente la caméra vers le point B. Par chance, des rotations de caméra n'induisent qu'une transformation homographique de l'image, qui peut être calculée facilement comme décrit plus haut. Dans les deux cas, un événement doit être généré (un "clic de souris") quand l'utilisateur considère que le curseur est précisément positionné par-dessus le point physique. Dans notre système, cet événement peut être généré en utilisant une touche de clavier, un bouton de téléphone en contexte mobile, une pression sur une zone d'un écran tactile, ou encore une commande vocale si des lunettes de RA sont utilisées. Plus généralement, trois touches seulement sont utilisées pour contrôler notre système : une pour "cliquer" ou "glisser - déposer", une pour annuler l'opération en cours ou demander une reprise de pose et une pour défiler dans le menu des outils.
Avant d'être en mesure de modéliser la scène, l'utilisateur doit calibrer la caméra : cela est réalisé en indiquant deux couples de droites parallèles entre elles dans la scène 3D, les droites du premier couple étant perpendiculaires à celles du deuxième (figure 5.10, capture a). Les points obtenus à l'intersection de ces quatre droites correspondent à la projection d'un rectangle, à partir de laquelle la focale de la caméra est estimée selon la méthode décrite dans [32] (le point principal est supposé situé au centre de l'image). L'utilisateur peut aussi déplacer l'origine du repère associé au modèle (dit *repère monde*), modifiant le rayon des lieux possible de ce point. La distance de ce point au centre optique est choisie de telle sorte que le vecteur unitaire vertical ait une certaine taille, fixée à l'avance, dans l'image. L'utilisateur peut enfin changer l'échelle de la scène en déplaçant l'extrémité de ce vecteur (au centre du carré bleu dans la capture a).
Une fois les paramètres intrinsèques de la caméra et l'échelle de la scène définis, l'utilisateur peut commencer à modéliser la scène. Les nouvelles faces sont instanciées au contact des faces existantes (intersection du rayon inverse avec la face rencontrée en premier le long du rayon) ou du sol (plan $z = 0$) par défaut. Les contacts sont garantis en "accrochant" des points de la scène qui apparaissent dans une couleur spécifique quand le curseur est suffisamment proche d'eux : jaune pour le sommet d'une face, bleu pour le milieu d'un segment, rouge pour le point le plus proche d'une segment, s'il ne s'agit pas du milieu. La face située sous le curseur est aussi soulignée de manière spécifique, et de nouveaux points peuvent être définis sur une face à l'intersection du rayon inverse. Enfin, la plupart des actions sont guidées par une droite inférée, qui peut prendre les couleurs des axes du repère monde (par exemple pour suggérer un déplacement vertical, le guide est affiché en bleu, couleur de l'axe vertical).
Six outils différents ont été implémentés, résumé en table 5.5. La figure 5.10 illustre comment une maquette de maison peut être modélisée en utilisant les outils **Ajouter**, **Pousser-Tirer**, **Couper** et **Déplacer**. Une face rectangulaire est **Ajoutée** en utilisant deux guides, AB et BC : conformément aux règles de priorité décrites au pied de la table 5.5, A est défini sur le sol et le segment AB suit l'axe rouge (capture b) ; le sommet C est déplacé dans un plan orthogonal à AB (capture c) et finalement sur la verticale de B (capture d) ; la nouvelle face est **Poussée - Tirée** de manière à former une boite. Un segment de droite est tracé pour **Couper** en deux parts égales la face supérieure de la boite (capture e), puis **Déplacé** le long de l'axe vertical de manière à former le toit de la maison (capture f).

5.3.4 Expérimentations

Les expériences décrites ci-dessous ont été réalisée sur un ordinateur portable Dell Precision M6300, couplé à une simple *webcam* Logitech. Le système tourne à cadence vidéo en mode standard, et à une fréquence de 2 à 6 images par seconde lorsque la procédure de ré-initialisation de la pose à partir des points de SIFT est appelée. Les vidéos 12, 13 et 14 montrent le système en action. Une scène miniature est utilisée pour mesurer la précision de la méthode (vidéo 12). Plusieurs phase de modélisation et de déplacement alternent durant la session de travail

	**Ajouter** une face rectangulaire en désignant trois points $A, B, C^{(*)}$. Le rectangle est généré dans le plan ABC tel que A et C soient opposés sur une diagonale.
	**Extraire** la texture et les descripteurs SIFT à l'intérieur de la face sélectionnée.
	**Pousser - Tirer** la face sélectionnée.
	Déplacer le sommet, le segment ou la face sélectionné(e).
	Couper la face sélectionnée en joignant deux de ses sommets.
	Supprimer la face sélectionnée.

(*) L'ordre de priorité concernant l'objet à sélectionner le long des rayons inverses est le suivant : pour le point A : sommet sélectionné > face sélectionnée > sol ; point le point B : sommet sélectionné > axe du repère monde parallèle à AB > face sur laquelle a été défini le point A ; pour le point C : sommet sélectionné > axe du repère monde parallèle à BC > plan orthogonal à AB.

TABLE 5.5 – Outils de modélisation utilisés dans notre système.

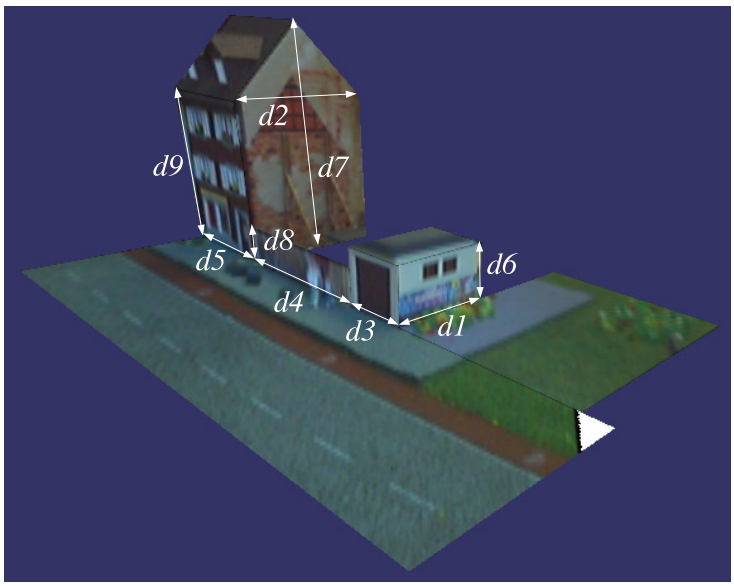
(figure 5.10). On peut facilement distinguer les phases de modélisation des phases de déplacement car la croix associée au curseur disparaît dans le second cas. La figure 5.11 montre les erreurs obtenues sur la géométrie de la scène (normalisées de telle sorte que la distance retrouvée d_1 corresponde à la distance attendue). Ces erreurs sont acceptables pour nombre d'applications : par exemple, afin de prouver que le modèle obtenu est utilisable pour la RA, nous avons ajouté un septième outil à notre interface de modélisation, permettant à l'utilisateur d'ajouter des objets virtuels sous le curseur. Ces objets semblent rigidement ancrés à la scène, y compris pour de larges amplitudes de distance à la scène (figure 5.10, captures g,h).
La phase de modélisation est par ailleurs illustrée sur une séquence prise en extérieur, montrant une piste d'athlétisme et un vestiaire (vidéo 13). La première partie de la vidéo montre comment l'ensemble initial de points de SIFT est étendu. Plusieurs rotations abruptes sont appliquées à la caméra, déclenchant la procédure de reprise. La caméra est ensuite calibrée et quelques opérations de modélisation sont effectuées. Enfin, un panneau virtuel est ajouté au bord de la piste (figure 5.12).
La vidéo 14 montre des résultats similaires. Cette vidéo est toutefois légèrement plus challengeante que la vidéo 13, dans la mesure où le sol seul est utilisé pour la calibration de la caméra, et un coureur à pied traverse le champ de vision de la caméra en fin de séquence, ce qui illustre la robustesse du système à la présence d'occultations partielles du modèle.

5.4 Conclusion et perspectives

Dans ce chapitre, nous avons présenté une méthode d'actualisation de la pose par suivi de plans texturés. Nous avons montré comment les problèmes de tremblements dus aux fluctuations du bruit sur les mesures pouvaient être évités à l'aide d'une sélection de modèle, et comment ce même outil pouvait être mis à profit pour distinguer les phases d'arrêt des phases de déplacement dans le cadre d'un système de modélisation *in situ*, permettant de court-circuiter le délai entre l'acquisition et l'exploitation des images utiles à une modélisation.
À la suite de ces travaux, nous avons proposé deux autres systèmes de modélisation *in situ*. Le premier, publié dans *The Visual Computer* 2011 [4], repose sur le suivi de deux régions planes (*blobs*) non parallèles (typiquement, un mur et le sol), désignées grossièrement en balayant la scène avec la caméra. Un filtrage particulaire est utilisé pour calculer la droite d'intersection entre les deux plans, en utilisant une mesure de vraisemblance géométrique (la droite comme lieu des points fixes de l'homologie $\mathbf{S} = \mathbf{H}_2^{-1}\mathbf{H}_1$ formée par les deux homographies) et une mesure de vraisemblance photométrique (la droite alignée avec des gradients forts de l'image). La convergence du filtre est validée visuellement par l'utilisateur, et la connaissance de la droite d'intersection permet de contraindre l'estimation de la géométrie des deux plans ainsi que du mouvement de la caméra, offrant des temps de calcul réduits et une meilleure précision que dans le cas non contraint. La figure 5.13 et surtout les vidéos 15 (scène d'intérieur illustrée en figure 5.13) et 16 (scène d'extérieur devant le Loria) permettent de mieux se rendre compte du fonctionnement de cette méthode. Elle est sans doute plus intéressante que la méthode présentée dans ce chapitre sur le plan mathématique, mais je la trouve finalement moins pertinente du

FIGURE 5.10 – Extraits d'une séquence de modélisation *in situ*.

point de vue utilisabilité. D'une part, la représentation de la scène en ensemble de blobs n'est pas directement exploitable pour la plupart des tâches mentionnées en introduction de la section 5.3.1.2. D'autre part, même si la détection fiable (puisque validée par l'utilisateur) de la droite d'intersection permet de contraindre le calcul de la géométrie bi-plane, ce calcul repose sur



	A(mm)	Δ(mm)	Δ/A(%)
d_1	61	Ref	Ref
d_2	88	-0.3	-0.3
d_3	38	1.5	3.9
d_4	92	4.2	4.6
d_5	75	-5.2	-6.9
d_6	34	-1.3	-3.8
d_7	139	-8.4	-6.0
d_8	21	0.8	3.8
d_9	95	-4.0	-4.2

FIGURE 5.11 – Géométrie retrouvée par modélisation *in situ* de la scène miniature, avec A la mesure attendue (mm), Δ l'erreur absolue de la mesure obtenue (mm) et Δ/A son erreur relative (%).

une géométrie binoculaire qui requiert un déplacement non nul entre les vues utilisées. Cela impose à l'utilisateur de se déplacer en permanence, comme avec le système proposé par Bunnun et al., et comme dans le SLAM visuel en général. À nouveau, cela n'est pas nécessairement problématique pour des petites scène, mais peut l'être pour des grands environnements (dans ces environnements, le SLAM visuel est généralement illustré à l'aide de séquences prises depuis une voiture ou un train, de sorte que le déplacement entre deux vues soit du même ordre que la distance à la scène). La méthode inspirée de Sketchup ne présente pas ce type de problème, grâce à l'approche incrémentale utilisée (chaque face est définie par rapport à une face précédemment définie, ou le sol par défaut).

En 2013, nous sommes revenus à une approche monoculaire, mais avons voulu cette fois évaluer l'apport d'un laser à un point et d'un capteur inertiel, intégrés au système de modélisation par l'image [17]. La précision de cette méthode n'était toutefois pas meilleure que celle de la méthode présentée en section 5.3. Le système était par ailleurs moins facile à utiliser et moins robuste, en raison notamment de la lenteur du capteur laser, des difficultés rencontrées pour le synchroniser avec les images vidéo, ainsi que de problèmes d'interférences entre le rayon du laser et la lumière du soleil en environnement extérieur. Il était aussi plus complexe à calibrer en raison des divers capteurs utilisés, devant tous "s'exprimer" dans le même repère (celui de la caméra).

La modélisation *in situ* n'a à ma connaissance pas fait l'objet de travaux significatifs depuis 2013. Les raisons sont sans doute multiples : (i) les systèmes existants sont pour la plupart assez complexes à utiliser, et nécessitent beaucoup d'ingénierie pour être implémentés ; (ii) ils sont par ailleurs globalement trop peu robustes ou imprécis : nos deux méthodes [20, 4] souffrent d'un problème de dérive, inhérent à l'odométrie visuelle. Un système utilisant des données capteurs, tel que celui décrit dans [PT01], ne présente pas de problème de dérive mais est trop imprécis pour obtenir des modèles de qualité ; (iii) l'arrivée des CNN a fait l'objet de toutes les attentions, favorisant l'élaboration de méthodes entièrement automatiques de reconstruction monoculaire. Des cartes de profondeurs, de même nature bien que plus approximatives que celles générées par une Kinect (autre "nouvel arrivant" ayant peut-être contribué à détourner l'attention des chercheurs du problème qui nous intéresse) peuvent être obtenues en utilisant des CNN [EF15]. Ces cartes peuvent être fusionnées à la volée, de manière à restituer la scène sous forme de volume de voxels [IKH+11]. Elles peuvent aussi être fusionnées avec des points obtenus par une technique de SLAM visuel, aboutissant à un maillage dense de la scène [MXS17]. Toutefois, ces volumes ou maillages ne sont pas représentés sous forme polyédrique comme cela est souhaitable dans de nombreuses applications. Ils présentent par ailleurs fréquemment des trous et autres artefacts.

FIGURE 5.12 – Image extraite de la première séquence d'extérieur (video 13).

Pour ces différentes raisons, un post-traitement manuel consistant à aligner des primitives élémentaires avec le nuage de point (ou autre représentation) est le plus souvent nécessaire. Ce travail s'avère en pratique au moins aussi fastidieux qu'une modélisation partant directement de l'image [5].
Pour ces raisons, je considère que les travaux autour de la modélisation *in situ* méritent d'être poursuivis, et que les principes fondamentaux de la méthode présentée dans ce chapitre (modélisation monoculaire, reposant sur des interactions simples et aboutissant à un modèle polyédrique) restent défendables. Plusieurs avancées récentes devraient permettre d'améliorer ce système, au prix d'une grande part d'ingénierie. Pour ce qui est de la partie calibration, nous avons proposé au chapitre 2 une méthode de détection de point de fuite automatique performante, qui pourrait facilement être incorporée au système actuel. Le problème de dérive de la pose peut être résolu à l'aide d'une approche corrective telle que celle présentée au chapitre 4. Quant aux opérations de modélisation, des cartes sémantiques et/ou de profondeur obtenues par CNN devraient pouvoir être utilisées pour mieux inférer et guider les opérations de l'utilisateur. Idéalement, on peut espérer aboutir à un système où les actions de l'utilisateur se cantonneraient à impulser puis valider ou invalider des propositions du système, et à se déplacer pour atteindre les différentes parties de la scène. Un ensemble restreint de tâches intuitives et faciles à réaliser, mais à même de fiabiliser grandement un système de reconstruction 3D.

5. Aligner des primitives élémentaires (boites, prismes, etc.) avec un nuage de points est par exemple réalisable en post-traitement avec des logiciels de *matchmoving* tels que Boujou© ou Voodoo camera tracker©. Certains de mes étudiants s'y sont essayé, avec beaucoup de peine !

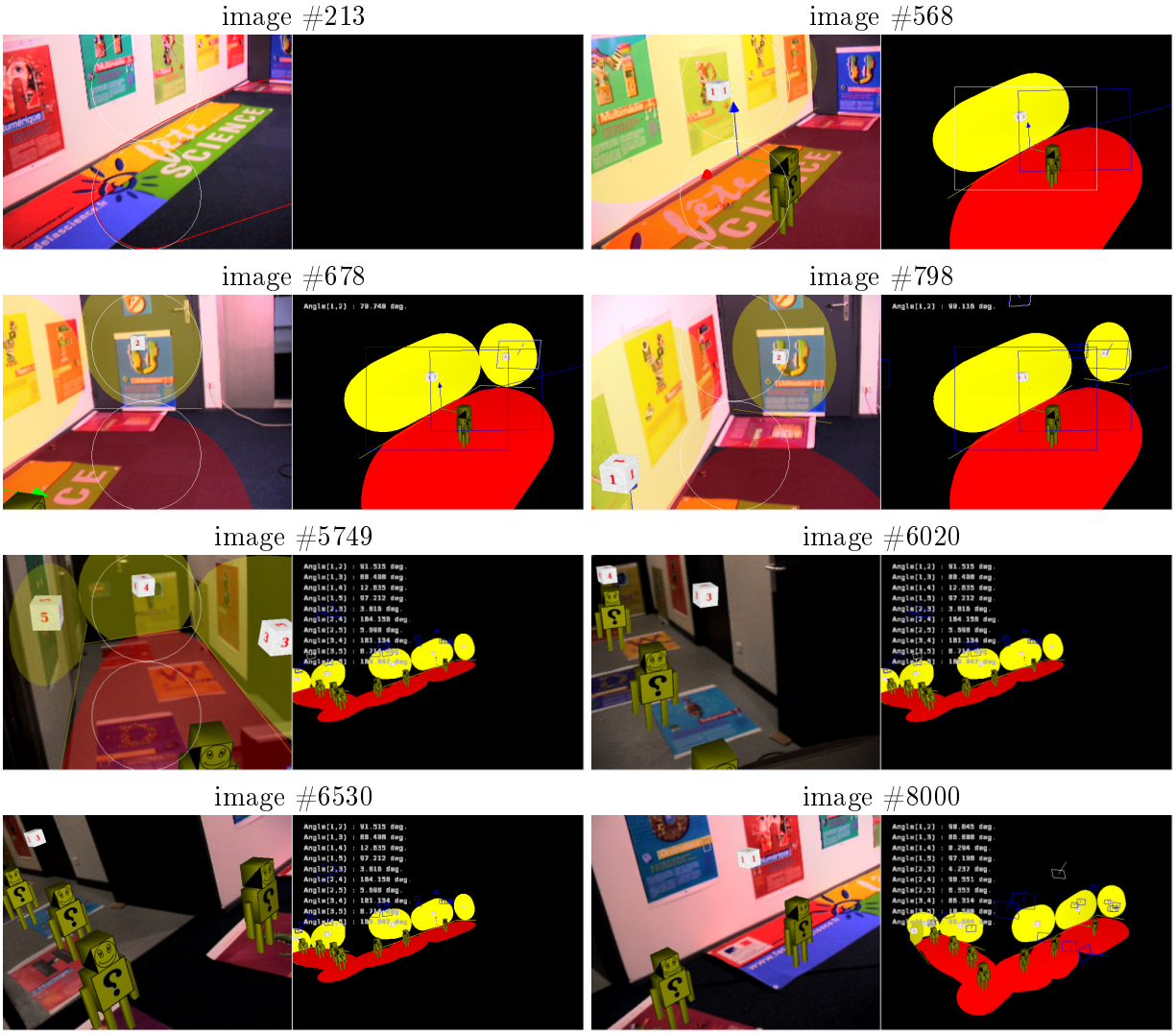


FIGURE 5.13 – Reconstruction à la volée d'un environnement intérieur utilisant l'interaction pour la définition des blobs et la validation [4].

Nous, figures, nous n'avons
Après tout qu'un vrai mérite,

C'est de simplifier le monde,
D'être un rêve qu'il se donne.

GUILLEVIC
Euclidiennes

Chapitre 6

Conclusion générale et recherches futures

FIGURE 6.1 – Paul Cézanne (1839-1906). *Pommes et oranges.*

Dans ce mémoire, nous nous sommes intéressé au calcul de pose en environnement bâti et j'ai choisi de livrer en fin de chaque chapitre les extensions me semblant en continuité naturelle avec le travail qui y était présenté. La prise en compte d'information sémantique a été déterminante dans les étapes de détection et reconnaissance de façade (chapitre 3), ainsi que dans le calcul de pose au moyen d'une approche corrective utilisant un modèle bayésien (chapitre 4). Elle est aussi évoquée comme piste prioritaire dans nos perspectives d'amélioration de notre méthode de détection de points de fuite (chapitre 2, prise en compte d'alignements de plus haut niveau) et de notre méthode de modélisation *in situ* (chapitre 5, utilisation de cartes sémantiques et/ou de profondeur pour mieux inférer et guider les opérations de l'utilisateur).
La prise en compte d'information sémantique me semble donc une composante incontournable de nos futurs travaux sur le positionnement visuel. Toutefois, si le problème du positionnement reste d'actualité dans le contexte des environnements bâtis (montée en puissance de la RA au service de la construction et de la géomatique, nouveaux enjeux liés à la conception de véhicules autonomes), de nouveaux de défis se posent dans d'autres domaines. Par exemple, d'après certains observateurs, une quatrième révolution industrielle est en cours, dans laquelle la RA semble avoir un rôle important à jouer. Mes futurs travaux ne se cantonneront donc pas aux applications en milieu bâti. Et, plutôt que de proposer de nouvelles méthodes pour chaque environnement spécifique, nous chercherons à généraliser le calcul de pose visuel à des scènes constituées d'objets quelconques, non nécessairemenent plans ni même polyédriques, et de surfaces potentiellement non texturées, lambertiennes ou spéculaires. La notion même d'objet isolé pourra ne pas être bien définie dans certaines scènes (voir par exemple la figure 6.2).

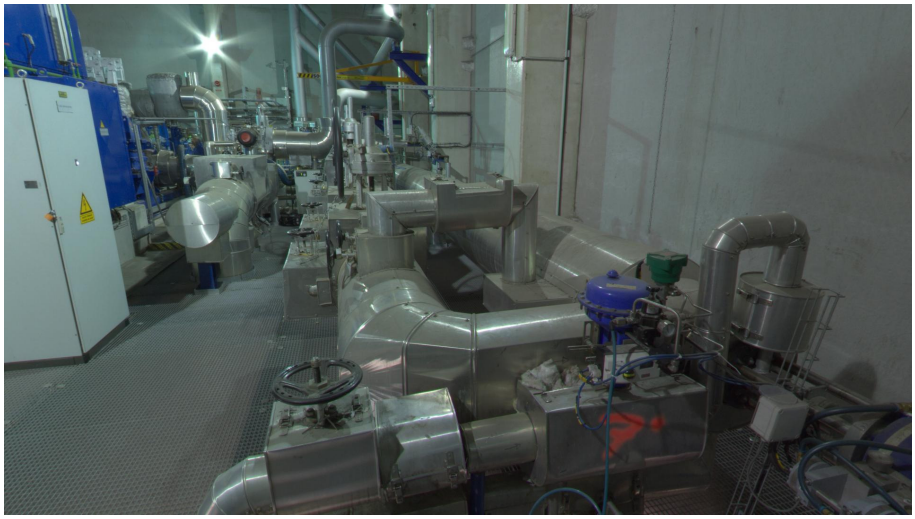


FIGURE 6.2 – Exemple de scène dans laquelle la notion d'objet isolé a peu de sens. Photographie prise par la société SBS-Interactive dans l'Usine d'Électricité de Metz (UEM).

Si une cartographie 3D des zones habitées (rues, marquages au sol des bâtiments, élévations, photographies etc.) a été entamée à l'échelle planétaire depuis de nombreuses années, cette démarche utile au positionnement urbain est loin de s'être généralisée à tous les domaines d'application de la RA. Aussi privilégierons-nous des approches n'impliquant pas, pour chaque nouvelle scène, une phase trop complexe d'adaptation au contexte, nécessitant par exemple d'acquérir un modèle géométrique précis et volumineux de la scène, ou un grand nombre d'images à taguer en vue de ré-entraîner un CNN. Proposer des méthodes de positionnement facilement adaptables à un nouveau contexte me semble être un enjeu crucial tant les besoins en RA se font sentir dans des secteurs de plus en plus variés, pour lesquels il existe rarement de solution "clé en main" ou même transférable à l'environnement et à la problématique visés. Dans le cadre de notre réflexion sur la généralisation du positionnement visuel à divers types d'environnements et d'objets, ce critère *d'adaptabilité* s'ajoute donc aux autres critères de performance déjà introduits au chapitre 1 (section 1.3(p19)).
En dehors des méthodes *ad hoc* utilisant la sémantique, les méthodes de positionnement pré-

sentées dans l'état de l'art de l'introduction générale (section 1.2(p13)) ne sont pas limitées aux environnements bâtis. Voyons toutefois à quelles difficultés supplémentaires ces méthodes peuvent être confrontées dans certains environnements, et comment nous pouvons les situer au regard du critère d'adaptabilité :

- les méthodes utilisant des descripteurs locaux sont relativement faciles à mettre en œuvre dans un nouvel environnement. Cela nécessite typiquement d'utiliser une méthode SFM ou SLAM pour obtenir un nuage de points 3D de la scène associé à des descripteurs locaux. De nombreux outils ergonomiques existent aujourd'hui pour réaliser cette tâche, qui demande typiquement des compétences d'infographiste. Ces méthodes peuvent toutefois échouer lorsque peu d'objets texturés sont présents dans la scène. Par ailleurs, les surfaces spéculaires peuvent conduire à apparier des points image (centres de tâches lumineuses) ne correspondant pas au même point physique. Ces problèmes s'ajoutent aux défauts déjà relevés de ces méthodes, considérées comme relativement instables (voir la section 1.2.1.1(p13)) ;
- les méthodes utilisant un descripteur global pour retrouver la pose par proximité de l'image avec une image de référence me paraissent moins faciles à adapter à de nouvelles scènes. Les descripteurs globaux sont adaptés à tout type d'environnement, mais la constitution d'une base de référence décrivant un nouvel environnement me semble problématique. Une solution SFM peut être utilisée, mais celle-ci doit être très dense en terme de points de vue représentés, ce qui me semble difficile à obtenir en pratique. On peut imaginer, comme dans [TAS+15], de densifier les images et le poses à l'aide de vues synthétiques, mais cette procédure repose dans [TAS+15] sur la disponibilité d'images panoramiques associées à des poses et des cartes de profondeur planes par morceaux disponibles dans Google Street View [1]. Obtenir ces éléments dans un autre contexte peut être très coûteux en terme de mesure ;
- même constat pour les méthodes de type PoseNet [KGC15]. Ces méthodes sont *a priori* capables de calculer des poses de niveau scène (à partir de l'image entière) ou de niveau objet (à partir d'une boite proposée) dans n'importe quel environnement. Toutefois, elles réclament tout comme les méthodes précédentes d'acquérir un grand nombre d'images de l'environnement ou de l'objet associées à des poses. Cette difficulté est amplifiée par le fait que le CNN doit être ré-entraîné à partir de ces données. Entraîner un CNN est une tâche relativement complexe requérant des compétences avancées en informatique. Une autre difficulté posée par PoseNet utilisé au niveau objet est que si plusieurs objets ont été reconnus dans l'image et que PoseNet infère une pose pour chacun d'eux, il n'est pas simple de fusionner ces poses, aucune mesure d'incertitude par exemple, ne leur étant associée ;
- le recalage 3D-2D a été envisagé avec des objets non texturés tels que des satellites [PMK11] ou des objets industriels métalliques [BCT+12]. Ces méthodes utilisent les contours d'un modèle de type CAO de l'objet recalé, éventuellement reconstruit à la volée à l'aide d'une caméra RGB-D [IKH+11]. Le problème se pose toutefois de l'initialisation de la pose, le GPS étant inopérant en intérieur. D'autres solutions dites *outside-in* existent pour les environnements intérieurs, telles que la triangulation par wifi ou l'utilisation de caméras externes filmant des leds fixées à la caméra. Ces dispositifs sont toutefois onéreux et complexes à calibrer, et leur présence dans certains lieux peut poser problème, voire être *a priori* exclue. D'autre part, le positionnement par recalage demande de posséder un modèle de type CAO de la scène, dont l'obtention peut également être très coûteuse, notamment dans les grands environnements (une technique telle que Kinect Fusion [IKH+11] souffre d'un problème de dérive, et par ailleurs les caméra RGB-D ne permettent pas de reconstruire des surfaces spéculaires). Ce modèle doit en outre être chargé en mémoire, sous une forme plus ou moins réduite, ce qui peut nuire au critère Cm de compacité du modèle introduit en section 1.3(p19).

Nous voyons ainsi que les méthodes les plus faciles à implanter dans un environnement quel-

1. `http://maps.google.com/help/maps/streetview/`

conque (approches utilisant des descripteurs locaux) sont aussi celles qui permettent le moins de bénéficier des dernières avancées de l'apprentissage profond. Notre réflexion portera donc sur la question suivante : *comment intégrer la prise en compte d'information sémantique à un système de positionnement visuel, sans que cela ne nuise à son adaptabilité ?* Nos pistes de réponses, détaillées ci-dessous, s'articulent autour de deux idées directrices, que nous pouvons résumer ainsi : favoriser l'abstraction géométrique des modèles de scène (section 6.1) et exploiter une information sémantique de classe, plutôt que d'instance (section 6.2). Parallèlement, nous réfléchirons (section 6.3) à des manières de simplifier l'acquisition des modèles et des données d'apprentissage utiles au positionnement visuel par l'une ou l'autre des méthodes décrites dans l'état de l'art du chapitre 1 ou envisagées ci-dessous.

6.1 Vers une abstraction géométrique de l'objet

6.1.1 Points abstraits

Afin de pallier les difficultés posées par la prise en compte de descripteurs locaux calculés sur des points "anonymes", les auteurs de [CRV+15] ont introduit la notion de "points de contrôle", désignés manuellement sur un maillage 3D de l'objet dont on cherche la pose, et détectés individuellement dans les images. Dans la méthode qu'ils proposent, un premier CNN est entraîné pour, étant donné un *patch* (glissant) dans l'image, donner la vraisemblance d'appartenance du *patch* à une des N classes représentant les points de contrôle. Par ailleurs, un CNN est entraîné spécifiquement pour chaque point de contrôle, afin de régresser la position du point étant donné un patch supposé le contenir. Ainsi, la notion de mise en correspondance de points anonymes est remplacée par celle de détection, ou reconnaissance, de points de contrôle bien identifiés. Cela permet d'espérer une meilleure précision des appariements utilisés dans un calcul PnP, la reconnaissance étant apprise à partir de nombreux exemples du même point physique vu sous des angles et conditions d'éclairage variés. Cette méthode convient à des objets texturés ou non, mais elle est difficilement adaptable à un nouveau contexte, du fait qu'un entraînement spécifique doit être réalisé pour chaque objet de la scène, utilisant un maillage 3D de l'objet (sur lequel est indiquée la position des points de contrôle) et un grand nombre d'images de cet objet associées à des poses (dans [CRV+15], 3000 vues/poses d'entraînement sont utilisées pour chaque objet considéré).
En revanche, une autre idée exploitée dans [CRV+15] me semble particulièrement intéressante dans le cadre de notre réflexion. Cette idée est que pour chaque point de contrôle physiquement attaché à l'objet, six autres points *abstraits* sont également "appris" (typiquement deux points de contrôle placés symétriquement autour du point physique dans chacune des trois directions orthogonales de l'espace, voir la figure 6.3). Après avoir entraîné le CNN, la position dans l'image des points de contrôle additionnels peut être obtenue par régression, de la même manière que pour les points physiques (plus précisément, les coordonnées des $6 + 1$ points de contrôle sont obtenues simultanément en sortie du même CNN, sous forme d'un vecteur à 2×7 dimensions).

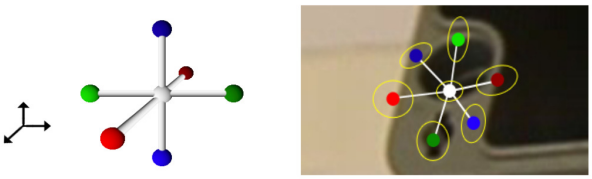


FIGURE 6.3 – Points de contrôle abstraits dans la méthode de Crivellaro et al. [CRV+15].

L'intérêt revendiqué de ces points abstraits, est que lorsqu'ils sont correctement détectés pour un seul point de contrôle physique donné, la pose peut être calculée par PnP à partir des 7 points ainsi reconnus, quand bien même aucun des autres points de contrôle physiques (et aucun de leurs voisins abstraits) n'ait été détecté. De mon point de vue, la portée de cette idée va au-delà de la robustesse aux occultations qu'elle implique. Les auteurs démontrent en effet, pour

la première fois me semble-t-il, qu'il est possible de reconnaître des points non physiques associés géométriquement à un objet. Cela n'avait rien d'évident, ces points pouvant être projetés sur (et donc prendre l'apparence de) différentes parties de l'objet ou de son arrière-plan, suivant le point de vue d'observation. Malheureusement, si la précision de la pose obtenue par cette méthode est supérieure à celle des méthode "state-of-the-art" en 2015, la précision de la détection des points abstraits n'est pas évaluée explicitement dans l'article.

6.1.2 Boites englobantes

Dans la méthode que nous venons de présenter, les points de contrôle doivent être désignés manuellement, relativement à un maillage 3D de l'objet, dont l'obtention peut s'avérer fastidieuse notamment pour des objets à géométrie complexe. Très récemment, les auteurs de [ORL18] ont proposé non plus d'apprendre à reconnaître des points physiques de l'objet, mais les sommets d'une boite englobant l'objet. Cette fois-ci, *tous* les points à détecter sont abstraits. Le problème des occultations est traité de la manière suivante : une fenêtre est détectée autour de l'objet et quadrillée en patchs. L'apprentissage du CNN vise à ce que n'importe lequel de ces patchs permette d'inférer, sous forme de carte de chaleur (*heatmap*), la position de l'un des sommets. Les heatmaps obtenues avec l'ensemble des patchs sont additionnées, et le maximum global de la somme est retenu comme étant la position du sommet recherché. De cette manière, si un nombre faible de patchs génère une heatmap incorrecte en raison de la présence d'une occultation à l'intérieur du patch (ou d'une confusion liée à la présence de symétries dans l'objet), la somme des heatmaps est sensée faire émerger la position correcte, celle-ci étant cumulée autant de fois qu'elle est correctement inférée par les autres patchs.
Grâce à cette approche, il n'est plus nécessaire de connaître un modèle précis de l'objet, mais uniquement une boite englobante. L'orientation de cette boite par rapport à l'objet, ainsi que ses dimensions peuvent d'ailleurs varier *a priori* sans nuire à la méthode, à condition de veiller à ce que les sommets de la boite ne soient pas trop éloignées de l'objet. Les heatmaps étant calculées à l'intérieur d'une fenêtre entourant l'objet, il est en effet important que les projections des sommets se retrouvent à l'intérieur de cette fenêtre (dans cette méthode, il n'est pas interdit que la fenêtre entourant l'objet soit confondue avec les bords de l'image, mais cela n'enlève pas la difficulté que les sommets de la boite doivent alors se projeter à l'intérieur de l'image).
Dans le même esprit que [ORL18] et la même année, les auteurs de [TSF18] proposent également de détecter dans l'image les points abstraits correspondant aux sommets d'une boite englobant l'objet. Leur solution diffère toutefois de celle de [ORL18], dans la mesure où les coordonnées 2D des sommets de la boite ne sont pas générés l'un après l'autre sous forme de heatmap, mais obtenus directement en sortie du CNN, sous forme d'un vecteur à 16 dimensions. Il s'agit donc d'une forme de régression neuronale comparable à celle opérée dans PoseNet, la sortie du CNN pouvant immédiatement être convertie en pose par résolution du problème P8P. Les avantages ici, par rapport à PoseNet, sont que la fonction de perte peut être définie de manière homogène en fonction de la distance entre les points obtenus et la vérité terrain, et qu'une seule pose peut être estimée lorsque plusieurs objets sont détectés simultanément, par simple résolution PnP sur l'ensemble des sommets détectés (à condition que ceux-ci soient exprimés dans le même repère). L'inconvénient est que le nombre de paramètres à estimer est plus important (16 au lieu de 6), ce qui n'est sans doute pas sans conséquence sur la taille de la base d'apprentissage et la diversité des exemples à fournir.

6.1.3 Ellipsoïdes englobantes

Cette dernière observation amène à s'interroger sur la pertinence, non pas de s'abstraire de la géométrie précise de l'objet et d'inférer des points 2D plutôt qu'une pose, mais du choix de la boite englobante comme modèle abstrait de l'objet. Des travaux récents ont montré qu'il est possible de résoudre le problème SFM et même SLAM au niveau objet, plutôt que point, en modélisant les objets par des ellipsoïdes [CRB16, NMS18, RCB18], et en appariant ces ellipsoïdes avec des ellipses extraites de l'image. Dans ces travaux, les ellipses sont obtenues à partir des fenêtres

de détection (générées par un CNN – ici YOLO, entraîné pour reconnaître les objets présents dans la scène). Malgré la simplicité de la méthode utilisée pour détecter les ellipses (leurs axes et dimensions sont simplement alignés avec ceux des fenêtres de détection de l'objet), des résultats cohérents géométriquement sont obtenus lorsque plusieurs objets sont considérés. Le cas d'un seul appariement est plus problématique, car la pose ne peut alors pas être obtenue de manière unique [10, 11, 50]. Il reste cependant qu'une ellipse n'a besoin que de 5 paramètres pour être définie. On peut ainsi envisager d'inférer ces paramètres à l'aide d'un CNN, dans l'espoir d'obtenir une extraction (et donc une pose) plus précise qu'en se contentant de l'ellipse inscrite dans la fenêtre de détection. Une architecture similaire à celle utilisée dans [TSF18] pourrait être utilisée, en remplaçant le vecteur à 16 dimensions par un vecteur à 5 dimensions, facilitant la procédure d'apprentissage (en fait, il vaudrait sans doute mieux utiliser 6 paramètres correspondant par exemple au centre et aux extrémités le long des demi-axes de l'ellipse, de manière à utiliser une fonction de perte homogène). Dans le cas où plusieurs objets sont détectés, le CNN peut être employé sur chaque fenêtre et la pose calculée à partir des n ellipses détectées. Lorsqu'un seul objet est détecté, plusieurs solutions de pose peuvent être générées mais on peut envisager d'utiliser d'autres informations pour lever l'ambiguïté (magnétomètres, points de fuite etc.).
L'intérêt d'utiliser des modèles géométriques abstraits, plutôt que détaillés, semble assez évident d'un point de vue adaptabilité. Nous verrons plus en détail en section 6.3 comment obtenir le plus facilement possible de tels modèles.

6.2 Sémantique de classe, plutôt que d'instance ?

Nous avons vu qu'un frein à l'implantation dans de nouveaux contextes d'un système de positionnement utilisant l'apprentissage profond, était la nécessité de ré-entraîner le reseau de neurones, à partir d'image et de poses acquises sur le nouveau site. Une manière radicale de simplifier cet apprentissage et ces acquisitions serait de faire en sorte que l'on puisse s'en passer... Tout du moins lorsqu'une nouvelle scène à prendre en compte correspond à un type (nous pourrions dire une classe) d'environnement "connu". Cela est tout à fait envisageable si la méthode de calcul de pose repose sur une sémantique de classe, plutôt que d'instance. Par exemple, notre méthode de recalage 3D-2D basée sur la sémantique présentée au chapitre 4 peut être utilisée dans n'importe quel lieu bâti, sans avoir à ré-entraîner le réseau calculant les cartes de sémantiques associées à ce lieu. Tout simplement parce que ce réseau a été entraîné à reconnaître des fenêtres, portes, etc. *en général* et non pas telle fenêtre ou telle porte d'un bâtiment particulier. L'inconvénient des approches [CRV+15, ORL18, TSF18] présentées en section 6.1, est que, pour tout nouvel objet, les points de contrôle doivent être re-définis, les données d'entraînement ré-acquises et le réseau ré-entraîné, quand bien même un objet du même type aurait déjà été appris. Les variations de matériau, de couleur et de géométrie du nouvel objet relativement à celui déjà appris rendent en effet caduque le réseau obtenu pour ce dernier.
Dans la réflexion qui suit, nous considérons qu'il existe des classes d'environnements (Usine, Navire, Bureau, Domicile, Ville, etc.) pouvant être définies comme un ensemble de classes d'objet rencontrées fréquemment dans ces environnements (par exemple, une usine comporte généralement des valves, des capteurs, des tuyaux, des machines, des armoires, etc., tout comme une scène urbaine comporte des fenêtres, des portes etc.). Si cette hypothèse est fondée, on peut concevoir d'élaborer un système de calcul de pose valable pour n'importe quelle instance d'une classe d'environnement donnée, sans avoir besoin d'apprendre à reconnaître les objets spécifiquement présents dans l'environnement, à condition que l'on y retrouve suffisamment d'instances des classes d'objets associées à la classe d'environnement. Évidemment, la tâche fastidieuse consistant à acquérir les images et poses utiles à l'apprentissage n'est pas réellement supprimée, mais déplacée du contexte spécifique vers un contexte plus général (utilisant des instances variées de chaque classe d'objet propre à la classe d'environnement), potentiellement agnostique du contexte spécifique. Le grand intérêt d'un tel glissement est que la collecte des données d'apprentissage d'une classe d'objet, ainsi que l'entraînement du CNN permettant de reconnaître n'importe quel instance de cette classe d'objet, peuvent être réalisés par des personnes plus au fait de la pro-

blématique (ce qui n'empêche pas de réfléchir à des outils permettant de leur faciliter la tâche) et ne doivent être réalisées qu'une seule fois, le CNN appris étant alors utilisable dans n'importe quel environnement instanciant la classe d'environnement (par exemple, dans n'importe quelle usine).
La sémantique de classe a jusqu'ici été très peu exploitée pour le calcul de pose, et essentiellement pour améliorer des descripteurs de points classiques (SemanticSIFT [AZ14]), valider des hypothèses de pose ([APV+15], voir la section 1.2.2.2(p19)), ou calculer des scores d'appariement en vue de favoriser les appariements les mieux notés dans un calcul de pose de type RANSAC [TSH+18]. Nous examinons plusieurs pistes pour mieux prendre en compte la sémantique de classe, qu'elle soit calculée au niveau image, au niveau objet ou niveau point.

6.2.1 Cartes sémantiques de classe

Dans [CWUF16] (voir la section 1.2.2.2(p19)), la sémantique est constitutive, avec d'autres critères, de la fonction d'énergie minimisée lors de l'inférence de pose. Cependant, cette inférence étant réalisée par recherche exhaustive dans un espace de paramètres discrétisé, la sémantique peut être vue comme mesure de validation d'hypothèses de pose, au même titre que dans [APV+15]. À ma connaissance, nos travaux présentés au chapitre 4 sont les seuls à calculer la pose par recalage itératif d'information sémantique (dense et susceptible "d'évoluer" à chaque itération de la méthode EM). Nous devrions pouvoir sans trop de difficultés adapter cette méthode à d'autres classes d'environnements, comprenant des objets non nécessairement plans. Le modèle de façade utilisé dans notre approche était uniquement constitué de mixtures de gaussiennes généralisées (une mixture par classe sémantique). Nous pourrions dans une approche plus générale considérer des mixtures de gaussiennes tridimensionnelles, "portées" par des ellipsoïdes, par exemple reconstruites automatiquement en utilisant la méthode [NMS18] mentionnée plus haut. Le principal intérêt de ce genre d'approche est que la phase d'utilisation ne requiert pas d'appariement explicite d'objets ou de points (donc pas de risque d'erreurs d'appariement), ni de détection fine (d'ellipses par exemple), étant donné qu'elle repose exclusivement sur la carte de sémantique et la description du modèle sous forme de mixtures de gaussiennes. En revanche, elle impose de connaître une pose approximative.

6.2.2 Classes d'objets

Une autre stratégie de calcul de pose utilisant la sémantique de classe peut être envisagée, qui ne requiert pas de pose initiale mais prend en compte des objets localisés. Historiquement, les CNN de reconnaissance d'objet ont été conçus, et le sont toujours aujourd'hui en grande majorité, pour reconnaître des classes d'objet, sous forme de fenêtres associées aux labels de classe et scores de reconnaissance. PASCAL VOC [EVGW+10], ImageNet [RDS+15] ou encore COCO [LMB+14] sont des exemples de jeux de données publics utilisés pour entraîner et comparer des réseaux de neurones convolutifs visant à détecter des classes d'objet sous cette forme. R-CNN [GDDM16], Fast R-CNN [Gir15], Faster R-CNN [RHGS15], YOLO [RDGF16] et SSD [LAE+16] sont des exemples de tels réseaux. Une méthode simple de calcul de pose utilisant la sémantique de classe (en fait, déjà présentée plus haut) pourrait donc consister à utiliser un réseau capable de reconnaître l'ensemble des classes d'objets constituant la classe d'environnement considérée puis, pour chaque environnement spécifique appartenant à cette classe :

- en phase préparatoire : acquérir une vidéo de la scène, reconnaître les objets de la classe d'environnement présents dans la vidéo et utiliser un algorithme SFM de niveau objet pour reconstruire la scène sous forme d'un nuage d'ellipsoïdes,
- en phase d'utilisation : reconnaître les classes d'objets présentes dans l'image et calculer la pose par PnP (ou plutôt PnE, "Perspective-n-Ellipsoids").

Ce schéma présente néanmoins deux difficultés. D'une part, il conduirait à des poses peu précises, si l'on s'en tient aux ellipses inscrites dans la fenêtre de détection. Et d'autre part, il présenterait un risque important d'erreur d'appariement lorsqu'une classe d'objet est représentée plusieurs fois dans la même image. Il n'y a dans ce cas pas d'autre alternative pour départager les hypothèses

d'appariement que d'utiliser un PnE robuste de type RANSAC, mais dont la performance est liée au nombre d'objets présents dans la scène, *a priori* plus faible au niveau objet qu'au niveau point.

6.2.3 Points de contrôle de classe

De nombreux travaux ont visé ces dernières années à détecter des points de contrôle de classes (souvent aussi appelés *landmarks*), particulièrement pour des visages [SWT13], des humains [TS14] ou du mobilier [WXL+16]. L'apprentissage du CNN dans ces travaux est cependant supervisé, c'est-à-dire qu'il est nécessaire de désigner manuellement des points de contrôle supposés être partagés par toutes les instances d'une même classe dans un grand nombre d'images utilisées comme vérité terrain pendant l'entraînement du réseau. Ce travail est non seulement fastidieux mais pose la difficulté de déterminer quels sont les points de contrôle qui, justement, sont communs à une classe d'objets. En 2017, un article a toutefois montré qu'il est possible d'obtenir des points de contrôle de classe de manière non supervisée [TBV17], la méthode conduisant d'ailleurs assez souvent, notamment dans le cas des visages, à des points similaires aux points de contrôle identifiés manuellement.
Nous pourrions exploiter les points de contrôle de classe en suivant le schéma suivant :

— en phase préparatoire : acquérir une vidéo de la scène, reconnaître les objets de la classe d'environnement présents dans la vidéo, pour chaque objet détecter les points de contrôle de classe de cet objet à l'intérieur de la fenêtre de détection et utiliser un algorithme SFM en considérant l'ensemble des points de contrôle détectés sur l'ensemble des objets pour reconstruire la scène sous forme d'un nuage de points,
— en phase d'utilisation : reconnaître les classes d'objets présentes dans l'image, détecter les points de contrôle associés à chaque objet et calculer la pose par PnP.

Ce schéma est très proche du précédent, les ellipsoïdes étant remplacées par des points. Il est aussi très proche du schéma classique SFM puis pose à partir d'un nuage de points associé à des descripteurs, ainsi que du schéma PnP à partir de points de contrôle désignés sur un modèle CAO [CRV+15]. Par rapport aux points de contrôle d'instance, les points de contrôle de classe présentent l'avantage de pouvoir être appris en amont pour toute une classe d'environnement. L'aspect non supervisé de l'apprentissage de ces points de contrôle est aussi très intéressant. En contrepartie, le risque d'obtenir des appariements incorrects est plus important avec les points de contrôle de classe qu'avec les points de contrôle d'instance, dans le cas ou des objets différents mais appartenant à la même classe d'objet sont détectés dans la même image. Le cas des points est cependant moins sensible que celui des ellipsoïdes, compte tenu du nombre plus important de primitives susceptible d'être obtenu, ce qui est favorable à la méthode RANSAC.
Nous venons de voir que l'étape préparatoire visant à acquérir des données d'entraînement et à effectivement entraîner un CNN à reconnaître des objets ou des points de contrôle dans un environnement spécifique peut être réalisée en amont, au niveau classe, sans aucune connaissance de l'environnement spécifique. On peut se demander s'il en va de même pour l'obtention du nuage de points 3D correspondant aux *points de contrôle* (de classe) susceptibles d'être détectés dans une vidéo de l'environnement spécifique. *A priori* la réponse semble être négative pour les deux raisons suivantes :

1. la position relative des objets dans l'environnement spécifique est propre à cet environnement,
2. même pour un objet isolé, si la sémantique des points de contrôle est la même pour toutes les instances d'objets d'une même classe, il n'en va pas de même de leur géométrie.

Pourtant, un article récent indique que la réponse à cette question pourrait être positive, en représentant la géométrie des classes d'objets par des modèles déformables [PZC+17]. Dans cet article, un CNN visant à détecter des points de contrôle d'une classe d'objet est appris de manière supervisée, à partir de données *ground truth* (points de contrôle positionnés sur des modèles CAO + positions des points de contrôle 2D dans de nombreuses vues des objets de la classe) issues du jeu de donnée PASCAL3D+ [XMS14]. Étant donnée une détection de points de contrôle dans

une image montrant un objet de la classe (ne faisant pas partie des données d'entraînement et dont il n'existe pas de modèle CAO), la méthode décrit comment reconstruire les points de contrôle et, simultanément, calculer la pose de la caméra (supposée calibrée) par rapport aux points reconstruits. Le principe est de construire un modèle déformable propre à la classe d'objet, en utilisant les modèles CAO et points de contrôle associés de tous les objets de la classe. Plus précisément, la matrice $\mathbf{S} \in \mathbb{R}^{3\times p}$ des p points de contrôle 3D se décompose en :

$$\mathbf{S} = \mathbf{B}_0 + \sum_{i=1}^{k} c_i \mathbf{B}_i, \tag{6.1}$$

où B_0 est la forme moyenne des objets de la classe et $\mathbf{B}_2, \ldots, \mathbf{B}_k$ sont plusieurs modes d'une possible variabilité de la forme, obtenus par ACP. Lorsque suffisamment de points de contrôle sont détectés, la pose peut être calculée en même temps que les coefficients $c_1, \ldots, c_k$ de la déformation.
Pour un nouvel objet instanciant la classe d'objets ainsi créée il suffirait donc, en théorie, d'une seule photographie (obtenue à l'aide d'une caméra calibrée) de cet objet pour détecter et reconstruire les points de contrôle associés, et obtenir en même temps la pose de la caméra correspondant à l'image. Si la photographie fait apparaître plusieurs objets distincts (d'une même classe ou de classes différentes), on obtient un unique nuage de points 3D contenant tous les points de contrôle de tous les objets, exprimés dans le repère de la caméra. Ce scénario paraît idéal, mais fonctionne uniquement sur des instances d'objets de formes très similaires, aux déformations près. Il faut par ailleurs souligner que les poses obtenues par cette méthode sont peu précises (voir les résultats présentés dans [PZC+17]). Enfin, le nombre de points à détecter pour calculer simultanément la pose et les points 3D n'est pas questionné dans l'article, mais ce nombre est nécessairement lié au nombre K de modes utilisés ($K = 2$ dans l'implémentation utilisée dans [PZC+17]).

6.2.4 Des primitives volumiques comme classes sémantiques universelles ?

Considérer des classes sémantiques d'objets ou de points permettrait de soulager le travail préparatoire requis pour adapter un système de calcul de pose à un nouvel environnement. Toutefois, la tâche consistant à apprendre l'apparence des objets ou des points observés sous différents angles reste à accomplir pour chaque nouvelle classe d'objet. De plus, un environnement peut comporter de nombreux objets non représentés par une classe apprise, ou des objets représentés par une classe mais non par les points de contrôle de cette classe. Une idée peut-être un peu hasardeuse, mais qui permettrait de prendre en compte n'importe quel objet ou classe d'objet en bénéficiant d'un apprentissage amont "universel" (indépendant de tout objet ou classe d'objet) serait de considérer des primitives volumétriques (cuboïdes, cylindres, ellipsoïdes etc.) comme briques de base du calcul de pose. Les peintres, observateurs aiguisés de la nature, savent bien que celle-ci peut être traitée "par le cylindre, la sphère, le cône" [2]. Les objets manufacturés, et en particulier ceux fabriqués à l'aide d'une machine à commande numérique ou d'une imprimante 3D sont fréquemment modélisés en amont sous forme d'arbres CSG [3][Lef13]. Les environnements urbains peuvent aussi facilement être modélisés sous forme d'assemblage de primitives volumiques, cette hypothèse ayant par exemple été utilisée pour concevoir le système de modélisation *in situ* présenté au chapitre 5.
L'approche par ellipsoïde englobante permet de bénéficier de la reconnaissance d'objet au niveau classe. En revanche, l'ellipsoïde englobante est une primitive à gros grain, pouvant apparaître en nombre réduit dans une image, ce qui peut avoir pour conséquence une faible précision de la pose et peu de robustesse dans son calcul (d'autant que deux objets différents d'une même classe

2. "Traitez la nature par le cylindre, la sphère, le cône, le tout mis en perspective, que chaque côté d'un objet, d'un plan, se dirige vers un point central". Lettre de Cézanne à Emile Bernard du 15 avril 1904.

3. Le standard CSG *Constructive Solid Geometry* définit un solide 3D sous forme d'une structure d'arbre, dont les feuilles sont des primitives volumiques (cuboïdes, cylindres, cones, etc.) et les noeuds des opérations booléennes entre les primitives (addition, intersection, etc.).

sont appariés indifféremment avec l'une ou l'autre des deux fenêtres détectées). À l'inverse, l'approche par points de contrôle présentée dans [CRV+15] permet d'obtenir un nombre de primitives suffisant pour un calcul de pose robuste et précis, mais requiert un apprentissage dédié à chaque point de contrôle, même si l'objet est reconnaissable au niveau classe indépendamment des points de contrôle. Les points de contrôle de niveau classe ne nécessitent pas d'apprentissage dédié, mais valent uniquement pour des objets très proches, au moins géométriquement, des objets de la classe considérée. Dans tous les cas, la reconnaissance des objets ou points de contrôle non apparentés à une classe doit être apprise spécifiquement, ce qui exige d'acquérir et d'annoter de nombreuses images en phase préparatoire. En revanche, si l'on suppose que l'on est capable d'identifier des primitives volumiques présentes dans une image, un calcul de pose utilisant un modèle CSG de la scène ou d'un objet de cette scène est envisageable sans avoir à ré-entraîner un CNN que ce soit au niveau instance ou au niveau classe. Le grain de ces primitives étant similaire à celui des points de contrôle, on peut espérer départager les hypothèses d'appariement multiples en se basant sur la cohérence du calcul PnP au sein d'une procédure de type RANSAC. Bien entendu, si des objets de la scène sont identifiables en amont sous forme de fenêtre de détection ou de carte sémantique, cette information peut être intégrée pour réduire la combinatoire du RANSAC. Une pose approximative pourrait être obtenue en se basant sur les centres des primitives volumiques appariés aux barycentres des régions détectées (*a priori* non confondus avec les projections des centres des primitives volumiques). Cette pose pourrait ensuite être affinée par un algorithme correctif utilisant les contours de l'image et des rendus synthétiques du modèle CSG.

FIGURE 6.4 – Exemples de cartes de profondeur obtenues par des CNN à partir de l'image uniquement. Images RGB en entrée du CNN (première ligne). Vérité terrain (seconde ligne). Résultats obtenus par la méthode [LRB+16] (troisième ligne). Résultats obtenus par la méthode [HWH18] (dernière ligne).

Le point important est que, si un CNN a été entraîné en amont à reconnaître des primitives volumiques (sur un grand nombre d'images issues de divers environnements), celui-ci devrait pouvoir opérer sur toute image présentant un ou plusieurs objets inédits (effectivement décomposables en primitives), sans avoir besoin de ré-entraîner le CNN ou d'associer l'objet à une classe. La question principale est cependant de déterminer si un tel apprentissage est plausible. Je n'ai pas de réponse certaine à cette question, mais des signes d'encouragement dans la littérature. Il me semble en effet que la détection de primitives volumiques est liée au problème d'estimer la carte de profondeur d'une image en utilisant un CNN. Ce problème est à l'étude depuis quelques années, et des résultats impressionnants (compte tenu de l'ambiguïté du problème et de la nature purement photométrique des informations sur lesquelles reposent ces méthodes) ont pu être obtenus [EPF14, EF15, LRB+16, HWH18] (voir par exemple les résultats de la figure 6.4). La détection de primitives volumiques est aussi liée à l'estimation des normales aux surfaces, autre problème qui se résout aujourd'hui assez bien avec des CNN, parfois conjointement avec l'estimation des profondeurs [EF15].

Au moins deux manières de désigner les primitives détectées sont envisageables : sous forme de fenêtres 2D supposées contenir les primitives (méthode de type *Selective Search* [UvdSGS13] + R-CNN [GDDM16]) ou sous forme d'une carte sémantique (méthode de type encodeur-décodeur [BHC15]). J'aurais tendance à privilégier la seconde forme, d'une part parce qu'elle offre une précision de détourage potentiellement supérieure à celle d'une fenêtre, et d'autre part en raison de la proximité du problème avec les problèmes mentionnés au paragraphe précédent. Nous pourrions d'ailleurs nous appuyer très étroitement sur l'architecture proposée dans [EF15], qui estime en même temps une carte de profondeur, une carte de normales et une carte de labels sémantiques (classes d'objets) sur des images de scènes d'intérieur. L'architecture du réseau repose sur le principe *coarse-to-fine* de l'architecture utilisée dans [EPF14] pour estimer la carte de profondeur uniquement. Dans cette architecture, un premier réseau est utilisé pour générer une carte de profondeur grossière, à basse résolution, correspondant à l'image. Les auteurs indiquent que ce premier réseau permet d'intégrer une compréhension globale de la scène utilisant les points de fuite, la localisation des objets et les alignements. Un second réseau est ensuite utilisé, pour affiner la carte de profondeur en tenant compte d'informations locales telles que les contours des objets et des murs. Ce deuxième réseau prend en entrée l'image originale et intègre le résultat du premier réseau, de manière à combiner l'information locale avec l'information globale. Le problème de détecter des primitives volumiques relève aussi me semble-t-il d'au moins deux niveaux d'échelle : au niveau grossier, les points de fuite devraient contribuer à détecter certaines primitives (boites, cylindres, ...), tandis que les contours et la couleur devraient aider à séparer les primitives ; au niveau fin, les silhouettes des primitives, ainsi que l'éclairage et la spécularité peuvent apporter des informations supplémentaires favorisant la reconnaissance des primitives volumiques.

L'apprentissage de ce réseau pourrait être faiblement supervisé. Il est facile en effet de générer des images de synthèse (et cartes de profondeurs et de normales) montrant des cuboïdes, des cylindres etc. sous différents angles, en faisant varier l'éclairage, l'arrière-plan, les matériaux, etc. et en introduisant des occultations. Des modèles CSG empruntés à différents secteurs d'activité (industrie, architecture, etc.) pourraient aussi être utilisés. Il serait sans doute utile aussi d'intégrer des photographies réelles d'objets modélisés en CSG et visibles dans leur environnement. Dans PASCAL3D+ [XMS14], des modèles CAO sont disponibles et associés à des images avec poses, mais un travail serait nécessaire pour transformer les modèles CAO (maillage 3D) en modèles CSG. Des solutions SFM pourraient aussi être exploitées, à condition de placer des primitives volumiques dans le nuage de points reconstruit (le logiciel gratuit Voodoo Camera Tracker [4] par exemple, permet à la fois de résoudre le problème SFM sur une vidéo et de placer des primitives volumiques au sein du nuage de points reconstruit). En outre, des méthodes automatiques permettant de générer un modèle CSG à partir d'un maillage dense sont actuellement à l'étude (voir la section 6.3.3 ci-dessous).

4. `https://www.viscoda.com/`

6.3 Acquisition des modèles et des données d'apprentissage

Une étape particulièrement contraignante dans la mise en œuvre des techniques modernes de calcul de pose est l'acquisition des modèles géométriques (ellipsoïdes, boites englobantes, modèles CAO) et des données d'entraînement utiles à ce calcul. Dans cette section, nous discutons de l'obtention de chacun des types de modèle utilisés, puis abordons la question de l'acquisition des données d'entraînement.

6.3.1 Acquisition des ellipsoïdes englobantes

Un des intérêts de choisir des ellipsoïdes comme primitives abstraites, en plus du faible nombre de paramètres utiles à leur description, est que celles-ci peuvent être reconstruites de manière automatique par SFM ou SLAM au niveau objet, comme démontré dans [CRB16, NMS18, RCB18]. Les ellipsoïdes reconstruites risquent toutefois d'être imprécises, du fait de considérer l'ellipse inscrite dans la fenêtre issue du CNN de reconnaissance d'objet. Nous avons commencé à regarder ce point avec Vincent Gaudillière, actuellement doctorant dans l'équipe Magrit, et nos premiers essais ont en effet révélé que, si les positions relatives des ellipsoïdes reconstruites sont généralement correctes, les dimensions des ellipsoïdes ne "collent" généralement pas très bien aux objets. Le terme "imprécises" est sans doute source de confusion car l'ellipsoïde étant un modèle abstrait de l'objet, il n'existe pas, sauf pour des objets de forme réellement ellipsoïdale, de vérité terrain dont on souhaite se rapprocher. Il est probable, mais nous devrons examiner plus avant cette question, que, de même qu'avec les boites englobantes, une certaine liberté existe dans la définition des dimensions et de l'orientation des axes de l'ellipsoïde, pour peu que la détection par CNN de l'ellipse correspondante soit apprise à partir des dimensions et orientations choisies. Toutefois, nos tests préliminaires ont montré qu'une méthode très simple (sans doute largement améliorable) de détection d'ellipse décrivant au plus près la répartition des segments dans la fenêtre de détection, permet d'obtenir une meilleure précision qu'à partir de la boite englobante, aussi bien de la forme des ellipsoïdes reconstruites que du calcul de pose reposant sur celles-ci. L'ellipsoïde est donc un objet très particulier, à la fois abstrait mais "presque" détectable en tant qu'objet physique sous forme d'ellipse. Cela n'est pas le cas des boites englobantes, qui ne sont pas détectables en tant que primitives géométriques 2D (la projection d'une boite 3D n'est pas une boite 2D). Ce que recouvre le terme "presque" reste à étudier, mais, étant donné ce qui a été dit dans le paragraphe précédent, il n'est pas forcément nécessaire que les ellipses collent parfaitement aux objets projetés. À l'inverse, si des ellipsoïdes mieux définies permettent d'améliorer la précision de la pose, il peut être intéressant de poursuivre nos recherches sur une détection plus précise des ellipses. Toutes ces questions seront examinées plus avant dans un avenir proche.

6.3.2 Acquisition des boites englobantes

La silhouette de la projection d'une boite 3D ne correspondant pas à une primitive géométrique 2D, il n'est pas possible, comme pour les ellipsoïdes, d'obtenir les boites englobantes utiles au calcul de pose tel que préconisé dans [ORL18, TSF18] par une technique de type SFM au niveau objet. Les boites englobantes peuvent toutefois être définies de différentes manières, par exemple en mesurant sur le site les dimensions de l'objet, ou encore en les positionnant manuellement autour de l'objet, à l'intérieur d'un nuage de points obtenu par SFM ou à l'aide d'une caméra RGB-D. Ces opérations ne sont pas très contraignantes d'un point de vue pratique, mais une difficulté me semble résider dans le choix de l'orientation des boites et de leurs dimensions, voire de leur nombre quand la scène est composée de multiples objets, en particulier lorsque certains objets sont proches et susceptible d'être regroupés au sein d'une même boite, ou au contraire lorsqu'un objet est composé de plusieurs parties occupant des volumes de tailles différentes. Considérer une unique boite englobant plusieurs objets simplifie les étapes de modélisation (une seule boite à définir, au lieu de n boites et $n-1$ transformations rigides entre ces boites) et d'apprentissage (1 apprentissage au lieu de n), mais risque de nuire à la détection des sommets

de cette boite dans les images, comme nous l'avons souligné dans la section précédente. De plus, fragmenter une boite en plusieurs sous-boites permet d'espérer une moindre sensibilité du calcul de pose à la présence d'occultations, tant que suffisamment de sous-boites ne sont pas occultées. Ainsi, le choix des boites n'est pas univoque et exige que l'opérateur soit conscient des considérations précédentes, ce qui n'est pas très favorable au critère d'adaptabilité.
Pour cette raison, il me semble préférable de travailler à l'élaboration d'une méthode automatique destinée à regrouper au sein de boites 3D les points d'un nuage donné. La difficulté réside dans la visée d'un compromis entre simplicité du modèle (le moins de boites possible), représentativité des points (le plus grand nombre de points doivent se trouver à l'intérieur d'une boite) et proximité des sommets des boites aux points. Difficulté d'autant plus grande que les objets représentés par les boites ne sont pas nécessairement (sont rarement) des boites, ce qui implique qu'il faille raisonner sur les volumes des boites plutôt que sur leurs faces. Le cadre *a contrario* me semble cependant particulièrement indiqué pour résoudre ce problème. Nous pourrions envisager par exemple de partitionner le volume du nuage de points en cubes subdivisés récursivement sous forme d'*octree* (aligné avec les directions principales du nuage de point), et repérer aux différentes échelles les cubes *significatifs* au sens défini dans le cadre *a contrario* [DMM07]. Et de la même manière que plusieurs intervalles contigus d'un histogramme peuvent se retrouver dans le même mode significatif maximal (voir l'annexe A), si le regroupement des intervalles est plus significatif que chaque intervalle isolément, les boites pourraient être formées de plusieurs cubes contigus spatialement.
Par ailleurs, la notion de contiguïté utilisée dans la détection des modes significatifs maximaux pourrait ne pas concerner uniquement la proximité spatiale au sein d'une même échelle, mais aussi la proximité d'échelle, dans le but d'obtenir des tailles de boites optimales. Ou bien, à l'inverse, nous pourrions vouloir obtenir des boites significatives à chaque échelle, entraîner des CNN à détecter les sommets de toutes les boites obtenues à toutes les échelles et, en phase d'utilisation, considérer uniquement les boites aux échelles observables depuis le point de vue de la caméra (approximé par exemple à l'aide de la pose précédemment calculée). Une telle stratégie est par exemple utilisée en infographie pour optimiser les performances d'un affichage par lancer de rayon de scènes représentées sous forme d'octrees de voxels [LK10]. Elle permettrait de résoudre élégamment le problème du niveau de détail, qui survient en calcul de pose dès lors que l'on considère de grands environnements à l'intérieur desquels on se déplace. Elle est d'ailleurs proche dans l'esprit de la technique pyramidale du MIP mapping que nous avons utilisée pour tenir compte du flou optique dans la méthode de suivi par synthèse de structure planes résumée en section 1.2.2.1(p16). Entraîner un CNN sur toutes les boites de toutes les échelles peut cependant être fastidieux et réclamer l'acquisition d'un grand nombre d'image autour de chaque objet et groupe d'objet à des distances variées. Nous abordons ce point à la fin de cette section.

6.3.3 Acquisition des modèles CSG

Si l'industrie 4.0 ou le processus BIM (*Building Information Modeling*) dans le domaine de la construction incitent à concevoir en amont les objets fabriqués ou les constructions sous une forme de type CSG, dans de nombreux scénarios il n'existe pas de modèle CSG de l'objet visé, mais un exemplaire physique dont il s'agit de générer le modèle CSG par rétro-conception, généralement à partir d'un scan 3D (un nuage de points) de l'objet. Ce problème est toutefois difficile, et malgré les nombreuses attentes en la matière ce n'est que très récemment que des résultats réellement intéressants (voir la figure 6.5) ont pu être obtenus [WXW18]. La méthode proposée cherche à minimiser (par une approche *bottom up* hautement combinatoire, incluant plusieurs étapes d'élagage d'hypothèses) une fonction d'énergie intégrant deux critères, l'un traduisant la précision du modèle (ce critère repose lui-même sur deux valeurs, le nombre de points attachés à chaque primitive, qui doit être grand et la distance des points aux primitives, qui doit être faible), l'autre la simplicité du modèle (tout simplement le nombre de primitives du modèle final, qui doit être faible). Un poids λ est utilisé pour équilibrer les deux critères selon l'objectif visé.
Ce type d'approche semble prometteur, et sans doute mieux adapté qu'une approche qui serait basée sur la méthodologie *a contrario*. Contrairement au cas des boites 3D, il faut en effet

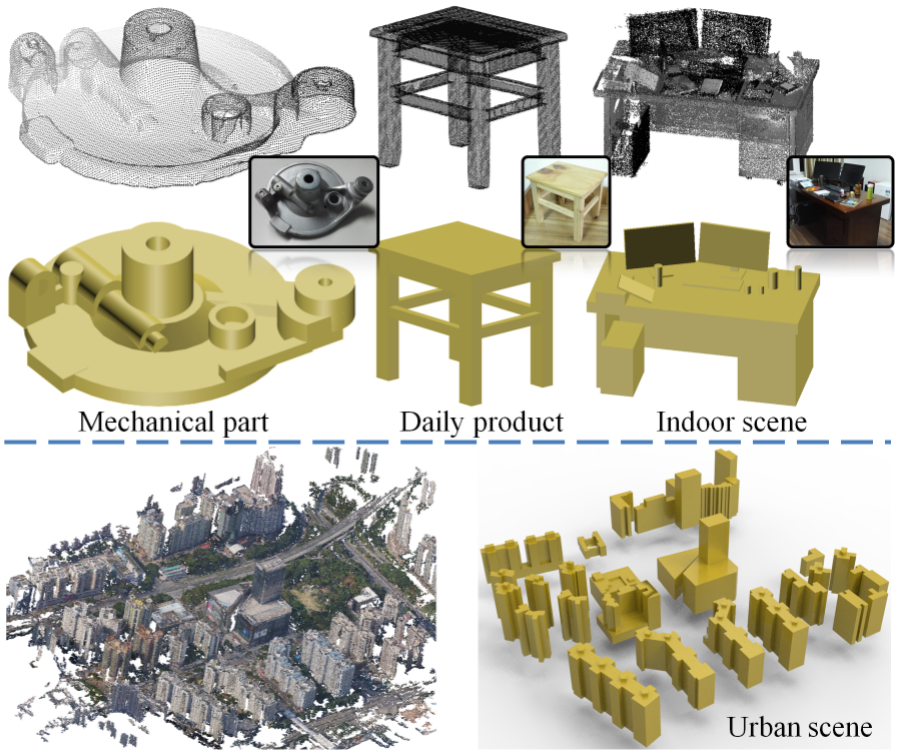


FIGURE 6.5 – Exemples de modèles CSG obtenus à partir de maillages denses avec la méthode présentée dans [WXW18].

raisonner ici sur la surface d'objets singuliers, non sur leur volume, alors que les méthodes *a contrario* recherchent des événements à l'intérieur de boites. Cependant, plusieurs avancées ont eu lieu ces dernières années en matière de reconnaissance de formes à l'intérieur d'un nuage de points [QCSKG17], et il peut être intéressant d'examiner si ces avancées ne seraient pas à même de réduire la combinatoire d'une méthode telle que celle présentée dans [WXW18]. Une autre possibilité serait de raisonner dans une ou plusieurs images de l'objet ou de la scène, et d'utiliser des techniques interactives aidées par exemple par la connaissance des points de fuite de l'image et d'une carte de sémantique de la scène (ce qui rejoint notre conclusion du chapitre 5). Bien entendu, une méthode de détection automatique de primitives volumiques telle que suggérée en section 6.2.4 serait très utile dans ce cadre.

6.3.4 Acquisition des données d'apprentissage

La plupart des méthodes de calcul de pose reposant sur un CNN (par exemple [CRV+15, KGC15, ORL18, TSF18]) utilisent des vues de l'objet ou de la scène, accompagnées des poses ayant donné lieu à ces vues, pour entraîner le CNN. L'acquisition de ces données pour un nouvel objet ou une nouvelle scène est une tâche pénalisante du point de vue de l'adaptabilité de ces méthodes. La manière la plus simple de procéder est sans doute d'acquérir une vidéo de la scène et d'utiliser une méthode SFM pour générer les poses associées aux images de la vidéo. Toutefois, une reconnaissance de points de contrôle ou un calcul de pose par régression CNN peuvent donner lieu à des résultats incorrects si les vues utilisées pour l'apprentissage ne "couvrent" pas suffisamment bien le sous-espace de $SE(3)$ susceptible d'être parcouru par l'opérateur en phase d'utilisation. Il s'agit d'un pré-requis fort, qui nécessite de se déplacer autour de chaque objet, de s'en approcher et de s'en éloigner sous plusieurs angles, sans garantie de couvrir suffisamment l'objet. De plus, la nécessité de devoir obtenir les poses correspondant aux images acquises durant cette prise de vue impose d'autres contraintes, propres à l'algorithme SFM (par exemple ne pas faire subir de rotation pure à la caméra). Il semble difficile de garder toutes ces contraintes à l'esprit pendant que l'on filme une scène ou un objet. Des outils pourraient être conçus pour guider l'acquisition

des images utiles à l'apprentissage, à l'aide d'un affichage en RA. Un exemple de tel système a été proposé il y a quelques années pour faciliter le SLAM d'objet texturés [PRD09]. Des flèches 3D sont affichées en RA afin de suggérer les mouvements à appliquer à la caméra pour obtenir des vues utiles à la reconstruction. L'élaboration de ce type d'outil nécessite toutefois d'analyser finement ce que l'on entend par "couverture" suffisante de l'objet, qui peut varier suivant le CNN utilisé et sa robustesse aux changements de pose entre les vues d'apprentissage et une vue test.

Remerciements

De nombreuses personnes ont collaboré directement ou indirectement aux avancées décrites dans ce mémoire et/ou contribué à faire émerger certaines des idées exposées dans mon projet de recherche.
En premier lieu Marie-Odile Berger, qui m'a fait l'honneur, vers la fin du siècle dernier, de m'attribuer le sujet de stage sur l'illumination par synthèse des ponts de Paris, alors que j'étais étudiant en troisième année de l'ÉSIAL (École supérieure d'informatique et applications de Lorraine, aujourd'hui Télécom Nancy). Nos réflexions partagées au jour le jour et sa constante détermination en faveur du projet commun ont été une source d'énergie importante, et le principal catalyseur de mes contributions propres.
J'ai aussi beaucoup appris de mon passage dans le groupe Visual Geometry de l'Université d'Oxford, où j'ai eu l'inestimable privilège de travailler avec "les deux Andrews" (Fitzgibbons et Zisserman). Puissé-je avoir été quelque peu imprégné, parmi leurs nombreux talents, de celui qui permet de rendre limpides des concepts en apparence compliqués.
J'ai par ailleurs eu le bonheur de coencadrer, avec Marie-Odile, des doctorants : Javier Flavio Vigueras Gomez, aujourd'hui enseignant-chercheur titulaire à temps plein à l'Universidad Autónoma de San Luis Potosí (Mexique), Antoine Fond qui, après un passage chez Blippar, a intégré la société Synthesia basée à Londres et Vincent Gaudillière, actuellement en thèse dans l'équipe Magrit (depuis 2016) ; des post-doctorants : Diego Ortin Trasobares (entre 2005 et 2006), Evren Imre (entre 2007 et 2008) et Cong Yang (entre 2016 et 2017) ; des ingénieurs : Michael Aron (entre 2003 et 2004), Christel Lénonet (ente 2010 et 2012), Benjamen Dexheimer (entre 2012 et 2015) et de nombreux stagiaires issus du master informatique ou d'écoles d'ingénieurs (Mines de Nancy, Centrale-Supelec Metz, Telecom Nancy) de l'Université de Lorraine.
J'ai collaboré étroitement avec des chercheurs et industriels co-auteurs d'articles ou participants à des projets communs, dont la liste serait trop longue à énumérer (les co-auteurs sont bien sûr mentionnés dans mes références bibliographiques, page 145).
J'ai pu enfin bénéficier d'un environnement de travail riche et stimulant intellectuellement, grâce notamment à des discussions quotidiennes avec mes collègues, devenus amis, de l'équipe Magrit.
Je remercie chaleureusement toutes ces personnes, ce mémoire n'aurait pas pu exister sans les précieux et fructueux échanges que nous avons eus.

Appendices

Modes significatifs maximaux d'un histogramme (chapitre 2)

Soit L le nombre de bins d'un histogramme. Pour chaque intervalle discret $[a,b]$ de l'histogramme, où a et b sont des entiers dans $\{1,\ldots,L\}$, soit $k(a,b)$ le nombre de données (parmi M) dont la valeur est représentée par un bin entre a et b et soit $p(a,b)$ la probabilité *a priori* qu'une donnée ait sa valeur dans un bin entre a et b. Le NFA de l'intervalle $[a,b]$ est le nombre attendu d'intervalles aussi significatifs que celui observé :

$$\mathrm{NFA}([a,b]) = \frac{L(L+1)}{2}\mathcal{B}(M,k(a,b),p(a,b)),$$

où $\mathcal{B}(n,k,p) = \sum_{j=k}^{n}\binom{n}{j}p^j(1-p)^{n-j}$ dénote la queue de la loi binomiale de paramètres n et p et $L(L+1)/2$ est le nombre total d'intervalles possibles. Un intervalle $[a,b]$ est dit ϵ-*significatif* si $\mathrm{NFA}([a,b]) \leq \epsilon$, c'est-à-dire

$$\mathcal{B}(M,k(a,b),p(a,b)) < \frac{2\epsilon}{L(L+1)}. \tag{A.1}$$

De même que dans [DMM07], nous utilisons l'approximation de grande déviation de la queue binomiale, basée sur *l'entropie relative* d'un intervalle :

$$H([a,b]) = \begin{cases} 0 & \text{si } r(a,b) \leq p(a,b) \\ r(a,b)\log\frac{r(a,b)}{p(a,b)} + (1-r(a,b))\log\frac{(1-r(a,b))}{(1-p(a,b))} & \text{sinon,} \end{cases}$$

où $r(a,b) = k(a,b)/M$ est la densité de données dont les valeurs sont comprises dans l'intervalle de bins $[a,b]$. Le théorème des grandes déviations de Camér établit que, pour $p(a,b) < r(a,b) < 1$ et pour de grandes valeurs de M, on a :

$$-\frac{1}{M}\log\mathcal{B}(M,k(a,b),p(a,b)) \approx H([a,b]). \tag{A.2}$$

Partant des équations (A.1) et (A.2), on dit qu'un intervalle $[a,b]$ est ϵ-*significatif au sens de la grande déviation* si son entropie relative $H([a,b])$ est telle que

$$H([a,b]) > \frac{1}{M}\log\frac{L(L+1)}{2\epsilon}.$$

Inversement, on dit qu'un intervalle $[a,b]$ est un *trou* ϵ-*significatif* si $r(a,b) < p(a,b)$ et

$$r(a,b)\log\frac{r(a,b)}{p(a,b)} + (1-r(a,b))\log\frac{(1-r(a,b))}{(1-p(a,b))} > \frac{1}{M}\log\frac{L(L+1)}{2\epsilon}.$$

Un intervalle est qualifié de *mode significatif* (MS) s'il est un intervalle significatif et qu'il ne contient pas de trou significatif. Finalement, un intervalle I est appelé *mode significatif maximal* (MSM) s'il s'agit d'un MS et que pour tout MS $J \subset I, H(J) \leq H(I)$ et pour tout MS $J \supsetneq I, H(J) < H(I)$.

Résolution analytique du système polynomial pour $p = 2$ (chapitre 4)

Pour $p = 2$, annuler les dérivées partielles de $\tilde{R}$ (équation (4.9)) conduit à résoudre un système polynomial d'une équation quadratique en s et deux équations linéaires en t_x et t_y. La solution de ce système est la suivante :

$$\begin{cases} s = \frac{-2a_2a_7a_8 + a_3a_5a_8 + a_4a_6a_7 \pm \sqrt{\Delta}}{4a_9a_7a_8} \\ t_x = \frac{-a_3 - 2a_5 s}{2a_7} \\ t_y = \frac{-a_4 - 2a_6 s}{2a_8} \end{cases} \tag{B.1}$$

avec

$$\begin{aligned} \Delta &= -16a_1a_7^2a_8^2a_9 + 4a_2^2a_7^2a_8^2 - 4a_2a_3a_5a_7a_8^2 - 4a_2a_4a_6a_7^2a_8 \\ &\quad + a_3^2a_5^2a_8^2 + 4a_3^2a_7a_8^2a_9 + 2a_3a_4a_5a_6a_7a_8 + a_4^2a_6^2a_7^2 + 4a_4^2a_7^2a_8a_9 \\ a_1 &= -\sum_{i,j,k_j} \beta_{i,j,k_j} \left(\frac{x_i^2}{\sigma_{k_j,x}} + \frac{y_i^2}{\sigma_{k_j,y}} \right) \quad a_2 = \sum_{i,j,k_j} \beta_{i,j,k_j} \left(\frac{x_i \mu_{k_j,x}}{\sigma_{k_j,x}} + \frac{y_i \mu_{k_j,y}}{\sigma_{k_j,y}} \right) \\ a_3 &= 2\sum_{i,j,k_j} \beta_{i,j,k_j} \frac{x_i}{\sigma_{k_j,x}} \quad a_4 = 2\sum_{i,j,k_j} \beta_{i,j,k_j} \frac{y_i}{\sigma_{k_j,y}} \quad a_5 = -\sum_{i,j,k_j} \beta_{i,j,k_j} \frac{\mu_{k_j,x}}{\sigma_{k_j,x}} \\ a_6 &= -\sum_{i,j,k_j} \beta_{i,j,k_j} \frac{\mu_{k_j,y}}{\sigma_{k_j,y}} \quad a_7 = -\sum_{i,j,k_j} \frac{\beta_{i,j,k_j}}{\sigma_{k_j,x}} \quad a_8 = -\sum_{i,j,k_j} \frac{\beta_{i,j,k_j}}{\sigma_{k_j,y}} \quad a_9 = 2\sum_{i,j,k_j} \beta_{i,j,k_j} \end{aligned} \tag{B.2}$$

Bibliographie

[ADF12] Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari. Measuring the objectness of image windows. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 34(11), November 2012.

[ADV03] A. Almansa, A. Desolneux, and S. Vamech. Vanishing point detection without any a priori information. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 25(4) :502–507, April 2003.

[Aka74] H. Akaike. A new look at the statistical model identification. *IEEE Trans Aut Ctrl*, 19(6) :716–723, 1974.

[ALJ+99] Ronald Azuma, Jong Weon Lee, Bolan Jiang, Jun Park, Suya You, and Ulrich Neumann. Tracking in unprepared environments for augmented reality systems. *Computers & Graphics*, 23(6) :787 – 793, 1999.

[APV+15] Clemens Arth, Christian Pirchheim, Jonathan Ventura, Dieter Schmalstieg, and Vincent Lepetit. Instant outdoor localization and SLAM initialization from 2.5d maps. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2015.

[AZ14] R. Arandjelović and A. Zisserman. Visual vocabulary with a semantic twist. In *Asian Conference on Computer Vision*, 2014.

[AZG+19] Luis Alonso, Yan Ryan Zhang, Arnaud Grignard, Ariel Noyman, Yasushi Sakai, Markus Elkatsha, Ronan Doorley, and Kent Larson. Data-driven, evidence-based simulation of urban dynamics. use case volpe. *Unifying Themes in Complex Systems IX*, 04 2019.

[BCT+12] B. Besbes, S. N. Collette, M. Tamaazousti, S. Bourgeois, and V. Gay-Bellile. An interactive augmented reality system : A prototype for industrial maintenance training applications. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 269–270, Nov 2012.

[BHC15] Vijay Badrinarayanan, Ankur Handa, and Roberto Cipolla. Segnet : A deep convolutional encoder-decoder architecture for r obust semantic pixel-wise labelling. *CoRR*, abs/1505.07293, 2015.

[BM01] Simon Baker and Iain Matthews. Equivalence and efficiency of image alignment algorithms. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages I–I. IEEE, 2001.

[BM04] Selim Benhimane and Ezio Malis. Real-time image-based tracking of planes using efficient second-order minimization. In *Proceedings of the International Conference on Intelligent Robots and Systems*, pages 943–948, 2004.

[BMC08] Pished Bunnun and Walterio W. Mayol-Cuevas. OutlinAR : an assisted interactive model building system with reduced computational effort. In *IEEE / ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 61–64, Los Alamitos, CA, USA, 2008. IEEE Computer Society.

[Boz87] H. Bozdogan. Model Selection and Akaike's Information Criterion (AIC) ; The General Theory and its Analytical Extensions . *Psychometrika*, 52(3) :345–370, 1987.

[BS98] K. Bubna and C.V. Stewart. Model selection and surface merging in reconstruction algorithms. In *IEEE International Conference on Computer Vision (ICCV)*, pages 895–902, 1998.

[BTJ15] Andrei Bursuc, Giorgos Tolias, and Hervé Jégou. Kernel Local Descriptors with Implicit Rotation Matching. In *ACM International Conference on Multimedia Retrieval*, ACM International Conference on Multimedia Retrieval, Shanghai, China, 2015.

[BTVG06b] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf : Speeded up robust features. *European Conference on Computer Vision (ECCV)*, pages 404–417, 2006.

[CCC+16] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, Jose Neira, Ian Reid, and John J. Leonard. Past, present, and future of simultaneous localization and mapping : Toward the robust-perception age. *Trans. Rob.*, 32(6) :1309–1332, December 2016.

[CHL05] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[CLSF10] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. Brief : Binary robust independent elementary features. *European Conference on Computer Vision (ECCV)*, pages 778–792, 2010.

[CRB16] M. Crocco, C. Rubino, and A. Del Bue. Structure from motion with objects. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 00, pages 4141–4149, June 2016.

[CRV+15] A. Crivellaro, M. Rad, Y. Verdie, K. M. Yi, P. Fua, and V. Lepetit. A novel representation of parts for accurate 3d object detection and tracking in monocular images. In *IEEE International Conference on Computer Vision (ICCV)*, volume 00, pages 4391–4399, Dec. 2015.

[CWUF16] Hang Chu, Shenlong Wang, Raquel Urtasun, and Sanja Fidler. Housecraft : Building houses from rental ads and street views. In *European Conference on Computer Vision (ECCV)*, 2016.

[DEE08] P. Denis, J. H. Elder, and F. J. Estrada. Efficient edge-based methods for estimating manhattan frames in urban imagery. In *European Conference on Computer Vision (ECCV)*, 2008.

[DM02] Andrew J. Davison and David W. Murray. Simultaneous localization and map-building using active vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 24 :865–880, 2002.

[DM10] Amaury Dame and Eric Marchand. Accurate real-time tracking using mutual information. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 47–56, 2010.

[DMM07] A. Desolneux, L. Moisan, and J.-M. Morel. *From Gestalt Theory to Image Analysis : A Probabilistic Approach*. Springer Publishing Company, Incorporated, 1st edition, 2007.

[DMR16] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Deep image homography estimation. *arXiv preprint arXiv :1606.03798*, 2016.

[DS15] Jingming Dong and Stefano Soatto. Domain-size pooling in local descriptors : DSP-SIFT. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5097–5106, 2015.

[DTM96] P. E. Debevec, C. J. Taylor, and J. Malik. Modeling and Rendering Architecture from Photographs. In *Proc. SIGGRAPH 96*, August 1996.

[EF15] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2650–2658, 2015.

[EPF14] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *NIPS*, 2014.

[EVGW+10] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision (IJCV)*, 88(2) :303–338, June 2010.

[FB81] Martin A. Fischler and Robert C. Bolles. Random sample consensus : A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6) :381–395, June 1981.

[FCNL13] Clement Farabet, Camille Couprie, Laurent Najman, and Yann LeCun. Learning hierarchical features for scene labeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 35(8) :1915–1929, 2013.

[Fon18] Antoine Fond. *Image-based localization in urban environment : application to augmented reality.* Theses, Université de Lorraine, April 2018.

[FRD10] Björn Fröhlich, Erik Rodner, and Joachim Denzler. A fast approach for pixelwise labeling of facade images. In *International Conference on Pattern Recognition (ICPR)*, 2010.

[FZ98] A. W. Fitzgibbon and A. Zisserman. Automatic camera recovery for closed or open image sequences. In Springer-Verlag, editor, *European Conference on Computer Vision (ECCV)*, pages 311–326, 1998.

[GDDM14] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 580–587, 2014.

[GDDM16] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Region-based convolutional networks for accurate object detection and segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 38(1) :142–158, January 2016.

[Gir15] Ross Girshick. Fast r-cnn. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1440–1448, Washington, DC, USA, 2015. IEEE Computer Society.

[GJMG17] Raghudeep Gadde, Varun Jampani, Renaud Marlet, and Peter Gehler. Efficient 2d and 3d facade segmentation using auto-context. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2017.

[GL94] Jean-Luc Gauvain and Chin-Hui Lee. Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains. *IEEE Transactions on Speech and Audio Processing*, 2(2) :291–298, 1994.

[GvGJMR12] R. Grompone von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall. LSD : a Line Segment Detector. *Image Processing On Line*, 2 :35–55, 2012.

[HB98] Gregory D Hager and Peter N Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 20(10) :1025–1039, 1998.

[HBS14] Jan Hosang, Rodrigo Benenson, and Bernt Schiele. How good are detection proposals, really ? In *British Machine Vision Conference (BMVC)*, 2014.

[HD16] B. Harwood and T. Drummond. Fanng : Fast approximate nearest neighbour graphs. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5713–5722, June 2016.

[HEH05] Derek Hoiem, Alexei A. Efros, and Martial Hebert. Automatic photo pop-up. In *ACM SIGGRAPH 2005 Papers*, pages 577–584, 2005.

[HS88] Chris Harris and Mike Stephens. A combined corner and edge detector. In *In Proc. of Fourth Alvey Vision Conference*, pages 147–151, 1988.

[HWH18] Lei He, Guanghui Wang, and Zhanyi Hu. Learning depth from single images with deep neural network embedding focal length. *IEEE Transactions on Image Processing*, 27 :4676–4689, 2018.

[HZ04] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision.* Cambridge University Press, second edition, 2004.

[HZRS14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *European Conference on Computer Vision (ECCV)*, pages 346–361, 2014.

[IKH+11] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, and Andrew Fitzgibbon. Kinectfusion : Real-time 3D reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, pages 559–568, New York, NY, USA, 2011. ACM.

[JD02] Frédéric Jurie and Michel Dhome. Real time robust template matching. In *British Machine Vision Conference (BMVC)*, pages 1–10, 2002.

[JDSP10] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3304–3311, June 2010.

[KAAA13] Serkan Kivrak, Gokhan Arslan, Aydin Akgun, and Volkan Arslan. Augmented reality system applications in construction project activities. In *International Symposium on Automation and Robotics in Construction (ISARC)*, 06 2013.

[Kan02] K. Kanatani. Model Selection for Geometric Inference. In *Proceedings of 5th Asian Conference on Computer Vision, Melbourne, Australia*, pages 23–25, 2002.

[KB99] Hirokazu Kato and Mark Billinghurst. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In *The 2nd International Workshop on Augmented Reality (IWAR 99)*, pages 85–94, 02 1999.

[KC17] Alex Kendall and Roberto Cipolla. Geometric loss functions for camera pose regression with deep learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6555–6564, 07 2017.

[KF04] Jeongtae Kim and Jeffrey A Fessler. Intensity-based image registration using robust correlation coefficients. *IEEE transactions on medical imaging*, 23(11) :1430–1444, 2004.

[KG11] Jakub Krolewski and Piotr Gawrysiak. The mobile personal augmented reality navigation system. In Tadeusz Czachórski, Stanisław Kozielski, and Urszula Stańczyk, editors, *Man-Machine Interactions 2*, pages 105–113, Berlin, Heidelberg, 2011.

[KGC15] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet : A convolutional network for real-time 6-dof camera relocalization. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2938–2946. IEEE Computer Society, 2015.

[KM09] G. Klein and D. Murray. Parallel tracking and mapping on a camera phone. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 83–86, Oct 2009.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[KZ02] J. Kosecka and W. Zhang. Video compass. In *European Conference on Computer Vision (ECCV)*, 2002.

[KZ05b] Jana Košecká and Wei Zhang. Extraction, matching, and pose recovery based on dominant rectangular structures. *Computer Vision and Image Understanding (CVIU)*, 100(3) :274–293, December 2005.

[LAE+16] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. Ssd : Single shot multibox detector. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *European Conference on Computer Vision (ECCV)*, pages 21–37. Springer International Publishing, 2016.

[Lef13] Sylvain Lefebvre. Icesl : a gpu accelerated csg modeler and slicer. In *Proceedings of AEFA'13, 18th European Forum on Additive Manufacturing*, 2013.

[LGvGRM14] J. Lezama, R. Grompone von Gioi, G. Randall, and J.-M. Morel. Finding vanishing points via point alignments in image primal and dual domains. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[LHK09] D. C. Lee, M. Hebert, and T. Kanade. Geometric reasoning for single image structure recovery. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[LK+81] Bruce D Lucas, Takeo Kanade, et al. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI)*, volume 2, pages 647–679. Vancouver, BC, Canada, 1981.

[LK10] Samuli Laine and Tero Karras. Efficient sparse voxel octrees. In *Proceedings of the 2010 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, I3D '10, pages 55–63, New York, NY, USA, 2010. ACM.

[LKT17] Anabel L. Kečkeš and Igor Tomicic. Augmented reality in tourism - research and applications overview. *Interdisciplinary Description of Complex Systems*, 15 :158–168, Jun 2017.

[LMB+14] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco : Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *European Conference on Computer Vision (ECCV)*, pages 740–755, Cham, 2014. Springer International Publishing.

[LMRvG15] J. Lezama, J. M. Morel, G. Randall, and R. G. v. Gioi. A Contrario 2D Point Alignment Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 37(3) :499–512, March 2015.

[Low99] D. G. Lowe. Object recognition from local scale-invariant features. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1150–1157, Los Alamitos, CA, 1999.

[Low04] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2) :91–110, November 2004.

[LRB+16] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. In *2016 Fourth International Conference on 3D Vision (3DV)*, 10 2016.

[LSH10] Yunpeng Li, Noah Snavely, and Daniel P Huttenlocher. Location recognition using prioritized feature matching. In *European Conference on Computer Vision (ECCV)*, pages 791–804. Springer, 2010.

[LSX+13] Y. Lu, D. Song, Y. Xu, A. G. A. Perera, and S. Oh. Automatic building exterior mapping using multilayer feature graphs. In *IEEE International Conference on Automation Science and Engineering (CASE)*, 2013.

[LWJ+18] Yi Li, Gu Wang, Xiangyang Ji, Yu Xiang, and Dieter Fox. Deepim : Deep iterative matching for 6d pose estimation. In *European Conference on Computer Vision (ECCV)*, 2018.

[MBM+16] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodolà, Jan Svoboda, and Michael M Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. *arXiv preprint arXiv :1611.08402*, 2016.

[MHV+01] David Mattes, David R Haynor, Hubert Vesselle, Thomas K Lewellen, and William Eubank. Nonrigid multimodality image registration. *Medical imaging*, 4322(1) :1609–1620, 2001.

[MK00] Chikara Matsunaga and Kenichi Kanatani. Calibration of a moving camera using a planar pattern : Optimal computation, reliability evaluation, and stabilization by model selection. In David Vernon, editor, *European Conference on Computer Vision (ECCV)*, pages 595–609, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg.

[MMWG12] Andelo Martinovic, Markus Mathias, Julien Weissenberg, and Luc J. Van Gool. A three-layered approach to facade parsing. In *European Conference on Computer Vision (ECCV)*, volume 7578, pages 416–429, 2012.

[MWK08] Branislav Micusík, Horst Wildenauer, and Jana Kosecka. Detection and matching of rectilinear structures. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[MXS17] Tomoyuki Mukasa, Jiu Xu, and Björn Stenger. 3D scene mesh from CNN depth predictions and sparse monocular SLAM. *IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 912–919, 2017.

[NMS18] Lachlan Nicholson, Michael Milford, and Niko Sunderhauf. Quadricslam : Dual quadrics as slam landmarks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.

[ORL18] Markus Oberweger, Mahdi Rad, and Vincent Lepetit. Making Deep Heatmaps Robust to Partial Occlusions for 3D Object Pose Estimation. In *European Conference on Computer Vision (ECCV)*, 2018.

[OSD05] Ji-Young Oh, Wolfgang Stuerzlinger, and John Danahy. Comparing SESAME and Sketching on Paper for Conceptual 3D Design. In *EUROGRAPHICS Workshop on Sketch-Based Interfaces and Modeling*, 2005.

[PCI+07] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, June 2007.

[PMK11] A. Petit, E. Marchand, and K. Kanani. Vision-based space autonomous rendezvous : A case study. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 619–624, Sep. 2011.

[PMV03] Josien PW Pluim, JB Antoine Maintz, and Max A Viergever. Mutual-information-based registration of medical images : a survey. *IEEE transactions on medical imaging*, 22(8) :986–1004, 2003.

[PRD09] Qi Pan, Gerhard Reitmayr, and Tom Drummond. Interactive model reconstruction with user guidance. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 209–210, Orlando, Florida, USA, 2009.

[PT01] Wayne Piekarski and Bruce H. Thomas. Tinmith-Metro : New Outdoor Techniques for Creating City Models with an Augmented Reality Wearable Computer. In *ISWC 2001*, pages 31–38, 2001.

[PZC+17] Georgios Pavlakos, Xiaowei Zhou, Aaron Chan, Konstantinos G Derpanis, and Kostas Daniilidis. 6-dof object pose from semantic keypoints. In *International Conference on Robotics and Automation (ICRA)*, 2017.

[QCSKG17] R Qi Charles, Hao Su, Mo Kaichun, and Leonidas Guibas. Pointnet : Deep learning on point sets for 3d classification and segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 77–85, 07 2017.

[RAS17] Ignacio Rocco, Relja Arandjelović, and Josef Sivic. Convolutional neural network architecture for geometric matching. *arXiv preprint arXiv :1703.05593*, 2017.

[RASC14] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf : An astounding baseline for recognition. *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 512–519, 2014.

[RC96] B Srinivasa Reddy and Biswanath N Chatterji. An fft-based technique for translation, rotation, and scale-invariant image registration. *IEEE Transactions on Image Processing*, 5(8) :1266–1271, 1996.

[RCB18] C. Rubino, M. Crocco, and A. Del Bue. 3d object localisation from multi-view image detections. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 40(6) :1281–1294, June 2018.

[RD06a] Gerhard Reitmayr and Tom Drummond. Going out : Robust model-based tracking for outdoor augmented reality. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 109–118, Santa Barbara, CA, USA, 2006. IEEE CS.

[RD06b] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *European Conference on Computer Vision (ECCV)*, pages 430–443, Berlin, Heidelberg, 2006. Springer-Verlag.

[RDGF16] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once : Unified, real-time object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788. IEEE Computer Society, 2016.

[RDS+15] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3) :211–252, 2015.

[RHGS15] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn : Towards real-time object detection with region proposal networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 91–99. Curran Associates, Inc., 2015.

[San04] L. Santalò. *Integral Geometry and Geometric Probability*. Cambridge University Press, 2004.

[Sch78] G. Schwarz. Estimating the Dimension of a Model. *The Annals of Statistics*, 6(2) :461–464, 1978.

[SDS+15] Niko Sünderhauf, Feras Dayoub, Sareh Shirazi, Ben Upcroft, and Michael Milford. On the performance of convnet features for place recognition. *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4297–4304, 2015.

[SHSP17] Johannes Lutz Schönberger, Hans Hardmeier, Torsten Sattler, and Marc Pollefeys. Comparative evaluation of hand-crafted and learned local features. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[SLK17] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Efficient & effective prioritized matching for large-scale image-based localization. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 39(9) :1744–1756, 2017.

[SMD+18] Martin Sundermeyer, Zoltan-Csaba Marton, Maximilian Durner, Manuel Brucker, and Rudolph Triebel. Implicit 3d orientation learning for 6d object detection from rgb images. In *European Conference on Computer Vision (ECCV)*, September 2018.

[SMT+18] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, Fredrik Kahl, and Tomas Pajdla. Benchmarking 6DOF Outdoor Visual Localization in

Changing Conditions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, United States, June 2018.

[SSJ+15] Niko Suenderhauf, Sareh Shirazi, Adam Jacobson, Feras Dayoub, Edward Pepperell, Ben Upcroft, and Michael Milford. Place recognition with convnet landmarks : Viewpoint-robust, con dition-robust, training-free. In *Proceedings of Robotics : Science and Systems*, Rome, Italy, July 2015.

[SSSC05] Rahunathan Smriti, D Stredney, P Schmalbrock, and BD Clymer. Image registration using rigid registration and maximization of mutual information. In *MMVR13. The 13th Annual Medicine Meets Virtual Reality Conference, Long Beach, CA*, page 74, 2005.

[SWT13] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deep convolutional network cascade for facial point detection. In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 3476–3483, Washington, DC, USA, 2013. IEEE Computer Society.

[SZ14] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[Tar09] J.-P. Tardif. Non-iterative approach for fast and accurate vanishing point detection. In *IEEE International Conference on Computer Vision (ICCV)*, 2009.

[TAS+15] Akihiko Torii, Relja Arandjelović, Josef Sivic, Masatoshi Okutomi, and Tomas Pajdla. 24/7 place recognition by view synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, United States, June 2015.

[TBKL12] E. Tretyak, O. Barinova, P. Kohli, and V. Lempitsky. Geometric image parsing in man-made environments. *International Journal of Computer Vision (IJCV)*, 97(3) :305–321, May 2012.

[TBV17] James Thewlis, Hakan Bilen, and Andrea Vedaldi. Unsupervised learning of object landmarks by factorized spatial embeddings. In *IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[TCP+18] Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, and Quoc V. Le. Mnasnet : Platform-aware neural architecture search for mobile. *CoRR*, abs/1807.11626, 2018.

[TFZ98] P. Torr, A. W. Fitzgibbon, and A. Zisserman. Maintaining multiple motion model hypotheses over many views to recover matching and structure. In *IEEE International Conference on Computer Vision (ICCV)*, pages 485–491, Jan 1998.

[TKS+13] Olivier Teboul, Iasonas Kokkinos, Loïc Simon, Panagiotis Koutsourakis, and Nikos Paragios. Parsing facades with shape grammars and reinforcement learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 35(7) :1744–1756, 2013.

[Tor97] P.H.S. Torr. An Assessment of Information Criteria for Motion Model Selection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 47–52, June 1997.

[Tos87] G. Toscani. *Systèmes de Calibration et Perception du Mouvement en Vision Artificielle*. PhD thesis, Paris 11, 1987.

[TS14] Alexander Toshev and Christian Szegedy. Deeppose : Human pose estimation via deep neural networks. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1653–1660, 2014.

[TSF18] Bugra Tekin, Sudipta N. Sinha, and Pascal Fua. Real-Time Seamless Single Shot 6D Object Pose Prediction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[TSH+18] Carl Toft, Erik Stenborg, Lars Hammarstrand, Lucas Brynte, Marc Pollefeys, Torsten Sattler, and Fredrik Kahl. Semantic match consistency for long-term visual localization. In *European Conference on Computer Vision (ECCV)*, September 2018.

[UvdSGS13] J.R.R. Uijlings, K.E.A. van de Sande, T. Gevers, and A.W.M. Smeulders. Selective search for object recognition. *International Journal of Computer Vision (IJCV)*, 2013.

[VARS14] Jonathan Ventura, Clemens Arth, Gerhard Reitmayr, and Dieter Schmalstieg. Global localization from monocular slam on a mobile phone. *IEEE Transactions on Visualization and Computer Graphics*, 20(4) :531–539, April 2014.

[vdHDT+07] Anton van den Hengel, Anthony Dick, Thorsten Thormählen, Ben Ward, and Philip H. S. Torr. VideoTrace : Rapid Interactive Scene Modelling from Video. In *ACM SIGGRAPH 2007 papers*, page 86, NY, USA, 2007.

[VG07] Javier Flavio Vigueras Gomez. *An augmented reality system based on planar structures : design and assessment*. Theses, Université Henri Poincaré - Nancy 1, January 2007.

[VJ01] Paul Viola and Michael Jones. Robust real-time object detection. In *International Journal of Computer Vision (IJCV)*, 2001.

[VWI97] Paul Viola and William M Wells III. Alignment by maximization of mutual information. *International Journal of Computer Vision (IJCV)*, 24(2) :137–154, 1997.

[VZ12] A. Vedaldi and A. Zisserman. Self-similar sketch. In *European Conference on Computer Vision (ECCV)*, 2012.

[WH12] H. Wildenauer and A. Hanbury. Robust camera self-calibration from monocular images of Manhattan worlds. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[WXL+16] Jiajun Wu, Tianfan Xue, Joseph J. Lim, Yuandong Tian, Joshua B. Tenenbaum, Antonio Torralba, and William T. Freeman. Single image 3d interpreter network. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *European Conference on Computer Vision (ECCV)*, pages 365–382, Cham, 2016. Springer International Publishing.

[WXW18] Q. Wu, K. Xu, and J. Wang. Constructing 3D CSG models from 3D raw point clouds. *Computer Graphics Forum*, 37(5) :221–232, August 2018.

[XMS14] Yu Xiang, Roozbeh Mottaghi, and Silvio Savarese. Beyond pascal : A benchmark for 3d object detection in the wild. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2014.

[XOH13] Y. Xu, S. Oh, and A. Hoogs. A minimum error vanishing point detection approach for uncalibrated monocular images of man-made environments. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[XXJH03] Xiao-Shan Gao, Xiao-Rong Hou, Jianliang Tang, and Hang-Fei Cheng. Complete solution classification for the perspective-three-point problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 25(8) :930–943, Aug 2003.

[YHQT12] Chao Yang, Tian Han, Long Quan, and Chiew-Lan Tai. Parsing facade with rank-one approximation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1720–1727, 2012.

[YTFLF16] Kwang Moo Yi, Eduard Trulls Fortuny, Vincent Lepetit, and Pascal Fua. Lift : Learned invariant feature transform. *European Conference on Computer Vision (ECCV)*, 9910 :17. 467–483, 2016.

[ZD14] C. Lawrence Zitnick and Piotr Dollár. Edge boxes : Locating object proposals from edges. In *European Conference on Computer Vision (ECCV)*, Zurich, Switzerland, September 2014.

[ZW05] Siavash Zokai and George Wolberg. Image registration using log-polar mappings for recovery of large-scale similarity and projective transformations. *IEEE Transactions on Image Processing*, 14(10) :1422–1434, 2005.

[ZWJ16] M. Zhai, S. Workman, and N. Jacobs. Detecting vanishing points using global image context in a non-manhattan world. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Publications de l'auteur

Ouvrage

[1] Gilles Simon and Julien Decollogne. *Intégrer images réelles et images 3D - Post-production et réalité augmentée*. Hors collection. Dunod, 2006. http ://www.dunod.com/.

Chapitre d'ouvrage

[2] Gilles Simon and Marie-Odile Berger. Réalité Augmentée et/ou Mixte. In Lucas, Laurent, Loscos, Céline, Remion, and Yannick, editors, *Vidéo 3D : Capture, traitement et diffusion*, Hermes Science - Traité IC2, série Signal et image. Hermes-Lavoisier, September 2013.

Actes de conférences

[3] Marie-Odile Berger, Erwan Kerrien, Gilles Simon, Antoine Tabbone, Laurent Wendling, and Brigitte Wrobel-Dautcourt, editors. *Actes des Journées Francophones des Jeunes Chercheurs en Vision par Ordinateur - ORASIS 2003*. INRIA, 2003.

Revues internationales

[4] Gilles Simon and Marie-Odile Berger. Interactive Building and Augmentation of Piecewise Planar Environments Using the Intersection Lines. *The Visual Computer*, 27(9) :827–841, February 2011.

[5] Michael Aron, Gilles Simon, and Marie-Odile Berger. Use of Inertial Sensors to Support Video Tracking. *Computer Animation and Virtual Worlds*, 18 :57–68, 2007.

[6] Gilles Simon and Marie-Odile Berger. Pose Estimation for Planar Structures. *IEEE Computer Graphics and Applications*, 22(6) :46–53, 2002. Article dans revue scientifique avec comité de lecture.

[7] Gilles Simon and Marie-Odile Berger. Des méthodes efficaces pour l'incrustation d'objets virtuels dans des séquences d'images. *Traitement du Signal*, 16(1) :31–46, 1999. Article dans revue scientifique avec comité de lecture.

[8] Marie-Odile Berger, Brigitte Wrobel-Dautcourt, Sylvain Petitjean, and Gilles Simon. Mixing Synthetic and Video Images of an Outdoor Urban Environment. *Machine Vision and Applications*, 11(3) :145–159, 1999. Article dans revue scientifique avec comité de lecture.

[9] Marie-Odile Berger, Christine Chevrier, and Gilles Simon. Compositing Computer and Video Image Sequences : Robust Algorithms for the Reconstruction of the Camera Parameters. *Computer Graphics Forum*, 15(3) :10, August 1996.

Conférences internationales

[10] Vincent Gaudillière, Gilles Simon, and Marie-Odile Berger. Camera Relocalization with Ellipsoidal Abstraction of Objects. In *ISMAR 2019 - 18th IEEE International Symposium on Mixed and Augmented Reality*, Beijing, China, October 2019.

[11] Vincent Gaudillière, Gilles Simon, and Marie-Odile Berger. Camera Pose Estimation with Semantic 3D Model. In *IROS 2019 - 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Macau, Macau SAR China, November 2019.

[12] Gilles Simon, Antoine Fond, and Marie-Odile Berger. A-Contrario Horizon-First Vanishing Point Detection Using Second-Order Grouping Laws. In *ECCV 2018 - European Conference on Computer Vision*, Munich, Germany, September 2018.

[13] Vincent Gaudillière, Gilles Simon, and Marie-Odile Berger. Region-based epipolar and planar geometry estimation in low-textured environment. In *ICIP 2018 - 25th IEEE International Conference on Image Processing, Oct 2018, Athens, Greece*, Athènes, Greece, October 2018.

[14] Antoine Fond, Marie-Odile Berger, and Gilles Simon. Facade Proposals for Urban Augmented Reality. In *ISMAR 2017 - 16th IEEE International Symposium on Mixed and Augmented Reality*, Nantes, France, October 2017.

[15] Gilles Simon, Antoine Fond, and Marie-Odile Berger. A Simple and Effective Method to Detect Orthogonal Vanishing Points in Uncalibrated Images of Man-Made Environments. In *Eurographics 2016*, Lisbon, Portugal, May 2016.

[16] Stéphanie Fleck, Gilles Simon, and Christian Bastien. AIBLE : An Inquiry-Based Augmented Reality Environment for Teaching Astronomical Phenomena. In *13th IEEE International Symposium on Mixed and Augmented Reality - ISMAR 2014*, Munich, Germany, September 2014.

[17] Christel Léonet, Gilles Simon, and Marie-Odile Berger. In-Situ Interactive Modeling Using a Single-Point Laser Rangefinder Coupled with a New Hybrid Orientation Tracker. In *12th IEEE International Symposium on Mixed and Augmented Reality - ISMAR 2013*, Adelaide, Australia, October 2013.

[18] Stéphanie Fleck and Gilles Simon. An Augmented Reality Environment for Astronomy Learning in Elementary Grades : An Exploratory Study. In *25ème conférence francophone sur l'Interaction Homme-Machine, IHM'13*, Bordeaux, France, November 2013. AFIHM, ACM.

[19] Gilles Simon. Tracking-by-Synthesis Using Point Features and Pyramidal Blurring. In *10th IEEE International Symposium on Mixed and Augmented Reality - ISMAR 2011*, Basel, Switzerland, October 2011.

[20] Gilles Simon. In-Situ 3D Sketching Using a Video Camera as an Interaction and Tracking Device. In *31st Annual Conference of the European Association for Computer Graphics - Eurographics 2010*, Norrköping, Sweden, May 2010.

[21] Srikrishna Bhat, Marie-Odile Berger, Gilles Simon, and Frédéric Sur. Transitive Closure based visual words for point matching in video sequence. In *20th International Conference on Pattern Recognition - ICPR 2010*, Istanbul, Turkey, August 2010.

[22] Gilles Simon. Immersive Image-Based Modeling of Polyhedral Scenes. In Gudrun Klinker, Hideo Saito, and Tobias Höllerer, editors, *8th IEEE/ACM International Symposium on Mixed and Augmented Reality - ISMAR 2009*, 8th IEEE International Symposium on Mixed and Augmented Reality - ISMAR 2009 - Science & Technology Proceedings, pages 215 – 216, Orlando, United States, October 2009. IEEE. ISBN : 978-1-4244-5390-0.

[23] Gilles Simon and Marie-Odile Berger. Detection of the Intersection Lines in Multiplanar Environments : Application to Real-Time Estimation of the Camera-Scene Geometry. In *19th International Conference on Pattern Recognition - ICPR 2008*, pages 1–4, Tampa, United States, December 2008. IEEE.

[24] Gilles Simon. Automatic Online Walls Detection for Immediate Use in AR Tasks. In *5th IEEE and ACM International Symposium on Mixed and Augmented Reality - ISMAR'06*, University of California at Santa Barbara, United States, October 2006.

[25] Javier-Flavio Vigueras, Gilles Simon, and Marie-Odile Berger. Calibration Errors in Augmented Reality : a Practical Study. In *4th IEEE and ACM International Symposium on Mixed and Augmented Reality - ISMAR'05*, Fourth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'05), pages 154–163, Vienna, Austria, October 2005. IEEE.

[26] Javier-Flavio Vigueras, Marie-Odile Berger, and Gilles Simon. On the Influence of Fixing the Principal Point in Frame-by-Frame Multiplanar Calibration. In *International Conference on Pattern Recognition - ICPR'2004*, page 4 p, Cambridge, Royaume Uni, 2004. Colloque avec actes et comité de lecture. internationale.

[27] Michael Aron, Gilles Simon, and Marie-Odile Berger. Handling uncertain sensor data in vision-based camera tracking. In *Third International Symposium on Mixed and Augmented Reality - ISMAR'04*, pages 58–67, Arlington, USA, 2004. IEEE and ACM. Colloque avec actes et comité de lecture. internationale.

[28] Javier-Flavio Vigueras-Gomez, Marie-Odile Berger, and Gilles Simon. Iterative Multi-Planar Camera Calibration : Improving stability using Model Selection. In Eurographics Association, editor, *Vision, Video and Graphics - VVG'03*, page 8 p, Bath, UK, 2003. Colloque avec actes et comité de lecture. internationale.

[29] Simon Gibson, Alan Chalmers, Gilles Simon, Javier-Flavio Vigueras-Gomez, Marie-Odile Berger, Didier Stricker, and Wolfram Kresse. Photorealistic Augmented Reality. In *Second IEEE and ACM International Symposium on Mixed and Augmented Reality - ISMAR'03*, page 3 p, Tokyo, Japon, October 2003. IEEE, ACM. Colloque sur invitation. internationale.

[30] Gilles Simon and Marie-Odile Berger. Reconstructing while registering : a novel approach for markerless augmented reality. In *International Symposium on Mixed and Augmented Reality - ISMAR'02*, page 10 p, Darmstadt, Germany, September 2002. Colloque avec actes et comité de lecture. internationale.

[31] Gilles Simon and Marie-Odile Berger. Real time registration of known or recovered multi-planar structures : application to AR. In *13th British Machine Vision Conference 2002 - BMVC'2002*, pages 567–576, Cardiff, United Kingdom, 2002. Colloque avec actes et comité de lecture. internationale.

[32] Gilles Simon, Andrew W. Fitzgibbon, and Andrew Zisserman. Markerless Tracking using Planar Structures in the Scene. In *Proc. International Symposium on Augmented Reality*, page 9 p, none, October 2000.

[33] Gilles Simon and Marie-Odile Berger. Registration with a Moving Zoom Lens Camera for Augmented Reality Applications. In *Proceedings of 6th European Conference on Computer Vision*, page 17 p, Trinity College Dublin, Ireland, June 2000. Colloque avec actes et comité de lecture. internationale.

[34] Gilles Simon, Vincent Lepetit, and Marie-Odile Berger. Registration methods for harmonious integration of real worlds and computer generated objets. In *Eurographics, Short Papers & Demos*, pages 53–55, Milan, Italy, 1999. Colloque avec actes et comité de lecture.

[35] Gilles Simon and Marie-Odile Berger. Registration with a Zoom Lens Camera for Augmented Reality Applications. In *Second International Workshop on Augmented Reality*, page 10 p, San Francisco, CA, October 1999. Colloque avec actes et comité de lecture.

[36] Gilles Simon, Vincent Lepetit, and Marie-Odile Berger. Computer Vision Methods for Registration : Mixing 3D Knowledge & 2D Correspondences for Accurate Image Composition. In *International Workshop on Augmented Reality*, page 15 p, San francisco, USA, 1998. Colloque avec actes et comité de lecture.

[37] Gilles Simon and Marie-Odile Berger. A Two-stage Robust Statistical Method for Temporal Registration from Features of Various Type. In *Proceedings of 6th International Conference on Computer Vision*, pages 261–266, Bombay, India, 1998. Colloque avec actes et comité de lecture.

[38] Marie-Odile Berger and Gilles Simon. Robust Image Composition Algorithms for Augmented Reality. In R. Chin and T.-C. Pong, editors, *Proceedings of Third Asian Conference on Computer Vision - ACCV'98*, volume 1352 of *Lecture notes in computer science*, pages 360–367, Hong Kong, China, 1998. Colloque avec actes et comité de lecture.

[39] Marie-Odile Berger, Christine Chevrier, and Gilles Simon. Compositing Computer and Video Image Sequences : Robust Algorithms for the Reconstruction of the Camera Parameters. *Computer Graphics Forum*, 15(3) :10, August 1996.

Conférences nationales

[40] Vincent Gaudillière, Gilles Simon, and Marie-Odile Berger. Estimation des géométries planaire et épipolaire en environnement faiblement texturé basée sur la mise en correspondance de régions. In *RFIAP 2018 - Congrès Reconnaissance des Formes, Image, Apprentissage et Perception*, Marne-la-Vallée, France, June 2018.

[41] Antoine Fond, Marie-Odile Berger, and Gilles Simon. Generation of facade hypotheses based on contextual and structural information. In *Reconnaissance des Formes et Intelligence Artificielle*, Clermont Ferrand, France, June 2016.

[42] Gilles Simon and Marie-Odile Berger. Reconstruction et augmentation simultanées de scènes planes par morceaux. In *16e congrès francophone AFRIF-AFIA Reconnaissance des Formes et Intelligence Artificielle - RFIA 2008*, Amiens, France, January 2008.

[43] Javier-Flavio Vigueras, Gilles Simon, and Marie-Odile Berger. Erreurs de calibration en réalité augmentée : une étude pratique. In *15ème congrès francophone Reconnaissance des Formes et Intelligence Artificielle - RFIA 2006*, Tours, France, January 2006. AFRIF / AFIA.

[44] Michael Aron, Gilles Simon, and Marie-Odile Berger. Utilisation d'un capteur inertiel comme aide au suivi basé vision. In *15ème congrès francophone Reconnaissance des Formes et Intelligence Artificielle - RFIA 2006*, Tours, France, January 2006. AFRIF / AFIA.

[45] Javier-Flavio Vigueras-Gomez, Marie-Odile Berger, and Gilles Simon. Calibration multiplanaire d'une caméra : augmenter la stabilité en utilisant la sélection de modèles. In *Journées Francophones des Jeunes Chercheurs en Vision par Ordinateur - ORASIS'2003*, pages 147–156, Gérardmer, France, 2003. LORIA, INRIA-Lorraine. Colloque avec actes et comité de lecture. nationale.

[46] Gilles Simon and Marie-Odile Berger. Recalage temporel d'une structure plane par morceaux : application à la Réalité Augmentée temps réel. In *13ème Congrès Francophone AFRIF-AFIA de Reconnaissance des Formes et Intelligence Artificielle - RFIA'2002*, page 8 p, Angers, France, January 2002. Colloque avec actes et comité de lecture. nationale.

[47] Gilles Simon and Marie-Odile Berger. Une méthode statistique robuste à deux niveaux pour le recalage temporel à partir de primitives de type différent. In *RFIA'98*, pages 183–192, Clermont-Ferrand (France), 1998. Colloque avec actes et comité de lecture.

Workshops

[48] Antoine Fond, Marie-Odile Berger, and Gilles Simon. Prior-based facade rectification for AR in urban environment. In *ISMAR workshop on Urban Augmented Reality*, Fukuoka, Japan, September 2015.

[49] Gilles Simon and Marie-Odile Berger. Registration Methods for Harmonious Integration of Real Worlds and Computer Generated Objects. In *Advanced Research Workshop on Confluence of Computer Vision and Computer Graphics*, page 3 p, Ljubljana, Slovenia, August 1999. Colloque avec actes et comité de lecture.

Rapports techniques

[50] Vincent Gaudillière, Gilles Simon, and Marie-Odile Berger. Perspective-12-Quadric : An analytical solution to the camera pose estimation problem from conic-quadric correspondences. working paper or preprint, March 2019.

[51] Gilles Simon and Marie-Odile Berger. A two-stage robust statistical method for temporal registration from features of various type. Research Report RR-3235, INRIA, 1997.

Logiciels

[52] G. Simon. >V< : a matlab tool for fast and accurate detection of vanishing points in uncalibrated images of man-made environments. `https://members.loria.fr/GSimon/v/`, 2018.

[53] G. Dexheimer B., Simon and Berger M.-O. Ltrack : an android platform to rigidly track a real object in real time using a cad model of this object., 2016.

[54] G. Simon and S. Fleck. AIBLE – AstroRA : interface tangible de réalité augmentée pour l'appréhension des phénomènes astronomiques en école primaire. Dépôt à l'Agence pour la Protection des Programmes, numéro IDDN.FR.001.140007.000.R.P.2015.000.31235, 2012.

[55] E. Kerrien, G. Simon, M-O. Berger, and V. Lepetit. RALIB : une bibliothèque logicielle pour le traitement d'images, l'imagerie médicale, et la réalité augmentée. Dépôt à l'Agence pour la Protection des Programmes, numéro IDDN.FR.001.100005.000.R.P.2004.000.10000, 2004.

Thèses et mémoires

[56] Gilles Simon. *Vers un système de réalité augmentée autonome.* Theses, Université Henri Poincaré - Nancy 1, December 1999.

[57] Gilles Simon. Détermination du point de vue à partir de l'observation d'un objet 3D dont le modèle est connu. Rapport de mémoire de diplôme d'étude approfondie, Université Henri Poincaré - Nancy 1, September 1995.

Vulgarisation scientifique

[58] Marie-Odile Berger and Gilles Simon. Réalité augmentée : entre mythes et réalités. *Interstices*, March 2016.

[59] Gilles Simon. La Réalité Augmentée, May 2013. Article publié dans le magazine de l'Académie Lorraine des Sciences.

Résumé. Mesurer en temps réel la pose d'une caméra relativement à des repères tridimensionnels identifiés dans une image vidéo est un, sinon le pilier fondamental de la réalité augmentée. Nous proposons de résoudre ce problème dans des environnements plans par morceaux, à l'aide de la vision par ordinateur. Nous montrons qu'un système de positionnement plus précis que le GPS, et par ailleurs plus stable, plus rapide et moins coûteux en mémoire que d'autres systèmes de positionnement visuel introduits dans la littérature, peut être obtenu en faisant coopérer : approche probabiliste et géométrie aléatoire (détection *a contrario* des points de fuite de l'image), apprentissage profond (proposition de boites contenant des façades, élaboration d'un descripteur de façades basé sur un réseau de neurones convolutifs), inférence bayésienne (recalage par espérance-maximisation d'un modèle géométrique et sémantique compact des façades identifiées) et sélection de modèle (analyse des mouvements de la caméra par suivi de plans texturés). Nous décrivons de plus une méthode de modélisation *in situ*, qui permet d'obtenir de manière fiable, de par leur confrontation immédiate à la réalité, des modèles 3D utiles au calcul de pose tel que nous l'envisageons.

Abstract. Measuring a camera pose with respect to three-dimensional landmarks identified in a video image is one, if not the fundamental pillar of augmented reality. We propose to solve this problem in multi-planar environments, using computer vision. We show that a positioning system more accurate than the GPS and more stable, faster and less expensive in memory than other visual positioning systems introduced into the literature, can be obtained by combining : probabilistic approach and random geometry (*a-contrario* detection of vanishing points), deep learning (proposal of boxes containing facades, designing of a facade descriptor based on a convolutional neural network), Bayesian inference (registration based on expectation-maximization of a compact geometric and semantic model of the identified facades) and model selection (analysis of the camera motion by tracking textured planar surfaces). We also describe an immersive, image-based modeling tool, aimed at reliably obtain 3D models useful for calculating the pose as understood in this report.